Algorithmic Analysis of Complex Audio Scenes

Dissertation

zur

Erlangung des Doktorgrads (Dr. rer. nat.)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Rolf Bardeli

aus

Berlin

Bonn, Juni 2008

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

Diese Dissertation ist auf dem Hochschulschriftenserver der ULB Bonn
`http://hss.ulb.uni-bonn.de/dissonline`
elektronisch publiziert.

1. Referent: Prof. Dr. Michael Clausen
2. Referent: Prof. Dr. Andreas Weber
Tag der Promotion: 11. September 2008
Erscheinungsjahr: 2008

Algorithmic Analysis of Complex Audio Scenes

Rolf Bardeli

Abstract

In this thesis, we examine the problem of algorithmic analysis of complex audio scenes with a special emphasis on natural audio scenes. One of the driving goals behind this work is to develop tools for monitoring the presence of animals in areas of interest based on their vocalisations. This task, which often occurs in the evaluation of nature conservation measures, leads to a number of subproblems in audio scene analysis.

In order to develop and evaluate pattern recognition algorithms for animal sounds, a representative collection of such sounds is necessary. Building such a collection is beyond the scope of a single researcher and we therefore use data from the Animal Sound Archive of the Humboldt University of Berlin. Although a large portion of well annotated recordings from this archive has been available in digital form, little infrastructure for searching and sharing this data has been available. We describe a distributed infrastructure for searching, sharing and annotating animal sound collections collaboratively, which we have developed in this context.

Although searching animal sound databases by metadata gives good results for many applications, annotating all occurences of a specific sound is beyond the scope of human annotators. Moreover, finding similar vocalisations to that of an example is not feasible by using only metadata. We therefore propose an algorithm for content-based similarity search in animal sound databases. Based on principles of image processing, we develop suitable features for the description of animal sounds. We enhance a concept for content-based multimedia retrieval by a ranking scheme which makes it an efficient tool for similarity search.

One of the main sources of complexity in natural audio scenes, and the most difficult problem for pattern recognition, is the large number of sound sources which are active at the same time. We therefore examine methods for source separation based on microphone arrays. In particular, we propose an algorithm for the extraction of simpler components from complex audio scenes based on a sound complexity measure.

Finally, we introduce pattern recognition algorithms for the vocalisations of a number of bird species. Some of these species are interesting for reasons of nature conservation, while one of the species serves as a prototype for song birds with strongly structured songs.

Keywords: multimedia information retrieval, computational bioacoustics, animal sounds, similarity search, source separation, sound archives

Danksagung / Acknowledgements

During the preparation of my thesis, a number of people have been helpful, in one way or the other, inspiring or just friendly and I wish to express my gratefulness to all of them. Without them this thesis would not have come into existence.

First of all, I would like to thank *Prof. Dr. Michael Clausen* and his workgroup, in which this thesis has been written and where I have enjoyed an atmosphere of extraordinary friendlyness and intellectual freedom. Most prominently, *PD Dr. Frank Kurth* has always been a helpful source of advice and cooperation. Our student *Daniel Wolff* has turned out to be a highly motivated and very helpful cooperator. Some special thanks are in store for those members of the group who have always been (sometimes forced by the fact of sharing an office with me) available for discussion and help, namely *PD Dr. Meinard Müller*, *Dr. Tido Röder* and *Christian Fremerey*. I have also enjoyed the company of and inspiring talks to other people from the work group and the computer science department, in particular, *David Damm*, *Sebastian Ewert*, *Frank Schmidt*, *Yoon-ha Chang* and *Richard Schmied*.

Special thanks go to *Prof. Dr. Andreas Weber* for accepting to act as second examiner.

The main inspiration for my thesis has been brought by the German Federal Agency for Nature Conservation (*Bundesamt für Naturschutz*, BfN). In particular the question *Can you do it for bird song?* during the presentation of a query-by-whistling system for music recognition has been a starting point for my research. A grant from BfN for the research and development project *Bioakustische Mustererkennung* (bioacoustical pattern recognition) has made possible the research in Chapter 5 and some of the research in Chapter 4. The research in Chapter 2 has been supported by a grant from the German Research Foundation (*Deutsche Forschungsgemeinschaft*, DFG) for the project *Informationsinfrastrukturen für netzbasierte Forschungskooperation in der Bioakustik* (information infrastructure for webbased cooperation in bioacoustics).

An especially important and fruitful cooperation for my thesis has been with the Animal Sound Archive (Tierstimmenarchiv) of Humboldt University Berlin. Their will not only to share their massive collection of animal sound recordings but also to engage in cooperative research in computational aspects of bioacoustics and its application to animal monitoring has been very helpful and inspiring. We were very lucky to find in *Dr. Karl-Heinz Frommolt*, the curator of the archive, a highly competent and proliferous collaborator with deep knowledge from bioacoustics and recording technology to computers. It has been a pleasure to cooperate with *Klaus-Henry Tauchert* in animal sound recognition. I would also like to thank *Andreas Gnensch*, *Martina Koch* and all the student workers at the Animal Sound Archive.

Beyond the academic world, my family, in particular my parents and sister, have been a most loving support.

I would especially like to thank *Dr. Virginia Lewerenz* for all of her support and much more.

Finally I'd like to thank all my friends and family for healthful doses of distraction and for their understanding for times of reduced communication.

This thesis is dedicated to the following people:

Irmgard Bardeli
* 5.7.1913 † 25.8.2005

Prof. Dr. Manfred Yoshua Brixius
* 11.6.1934 † 28.3.2007

Klaus Schliemann
* 6.9.1972 † 12.6.2007

Contents

Chapter 1

Complex Audio Scenes: Introduction and Overview

Every day, we encounter complex audio scenes and, most of the time, we have little problem interpreting their contents. Our auditory senses are so well developed that we succeed in spectacular tasks. In crowds with a high number of speakers, we can concentrate on the one person we are interested in. At crossroads, we can distinguish the sounds of different cars approaching and we can also tell from which direction the sounds are coming. Close your eyes, and you can still get a fairly accurate mental image of which sound sources are present, where they are, and what is their cause. You even recognise when an unknown sound is present and, although you might wonder what it is, usually, you can still tell where it is coming from.

When we think of complex audio scenes, there is one example everybody is familiar with: natural audio scenes. Everybody has encountered, on one day or the other, the vast beauty and complexity of bird choirs in the morning. The conservation of nature — not only for the sake of its beauty — is one of the most important topics of our time. Astonishingly, the algorithmic analysis of complex audio scenes can be of great use in this context.

Measures for nature conservation are meaningless without methods for their evaluation. Often, assessing changes in population sizes of certain indicator species is the method of choice for this task. At the 1992 Earth Summit in Rio de Janeiro, the majority of governments have adopted the Convention on Biological Diversity. One of its central goals lies in the conservation of biological diversity. At the national level, it comes with the goal of "identifying and monitoring the important components of biological diversity that need to be conserved." Each participating country has to find an implementation of this goal. Up to now, monitoring the change in population sizes of animal species is performed by a large number of volunteers. Every volunteer assesses a fixed area a few times during each year. This method comes with a number of problems. First, the volunteers have to be well-trained in the recognition of animals and their voices. Second, results are not comparable between different persons and thus a comparison of results between years is only possible if the volunteer responsible for an area does not change. Third, regions that are difficult to access or should not be disturbed by the presence of humans can hardly be assessed by this method. Finally, certain times, such as nights, are usually not covered by human observers. The development of algorithms for the recognition of animal sounds from recordings of natural audio scenes could be a very helpful additional tool in an implementation of national monitoring programmes overcoming some of the problems inherent in conventional monitoring methods.

In this context, as well as in others, it is highly desirable, to recreate some of the capabilities of our auditory system in technical systems and applications in numerous fields can profit from such developments. For example, people wearing hearing aids usually report limitations to their ability to focus on audio sources of interest and are overwhelmed by the complexity of the audio scenes they encounter. As another example, good results in automatic speech recognition are only found in environments with as little noise as possible. Often, continuous recordings are made for the surveillance of certain areas. Manually analysing such recordings is a very exhausting and time-consuming task and algorithms for automatically or semi-automatically annotating such recordings would be very helpful.

The analysis of complex audio scenes is a broad field, concerning the multitude of possible audio scenes and audio sources, the variety of conceivable algorithmic tasks such as source separation and pattern recognition, as well as more technical topics such as the choice of recording equipment and its placement. Therefore, focusing is as important in research as it is in hearing.

Motivated by the problems in monitoring nature introduced above, in this thesis, we focus on natural audio scenes and the sound sources of interest will be animals, especially birds. In addition to these practical reasons, natural audio scenes are examined because they tend to be among the most complex audio scenes encountered.

In this introductory chapter, we first give an example illustrating the effects leading to high complexity in natural audio scenes and, hence, to difficulties in their algorithmic analysis. Afterwards, we will give an overview of the topics covered in this thesis.

Complex Audio Scenes

We will now discuss several effects occurring in natural audio scenes that contribute to their complexity. By starting with a simple scene and adding more and more of these effects, we will finally arrive at a typical complex audio scene. Most of the effects discussed here are independent of each other, thus constituting different *dimensions* of complexity. How much each of these effects affects an audio scene is strongly dependent of the recording location.

First, we start with only one audio source which is recorded in an anechoic chamber. In this way, the recording is not affected by the acoustics of the recording environment. Figure 1.1a shows the spectrogram of the barking of a dog. Already, there is some complexity in the signal. On the one hand, the barking is a complex modulated signal. On the other hand, calls and songs from animals show different levels of variability, depending on the species. Our example shows the spectrogram of two quite different barks from the same dog.

Now, we add effects stemming from the interaction of sound with the environment in which it is recorded. Let us assume that we are recording in a forest. Here, the propagation of the source signal to the receiver is influenced by trees. Some obstacles in the path to the microphone may attenuate the signal, while others not lying on the direct path of propagation can reflect the signal and create echoes. Each interaction of the signal with an obstacle is non-trivial and leads to attenuation and time-delay. Both of these effects are frequency-dependent. In Figure 1.1b, we sketch a hypothetical recording environment and show how the spectrum of the signal from Figure 1.1a is changed by the environmental interaction.

In natural audio scenes, multiple sources are active at most times and difficulties arise from the fact that they often overlap in time and frequency. Figure 1.1c shows the spectrogram of our hypothetical audio scene. In addition to the repeated barking of a dog, the song of

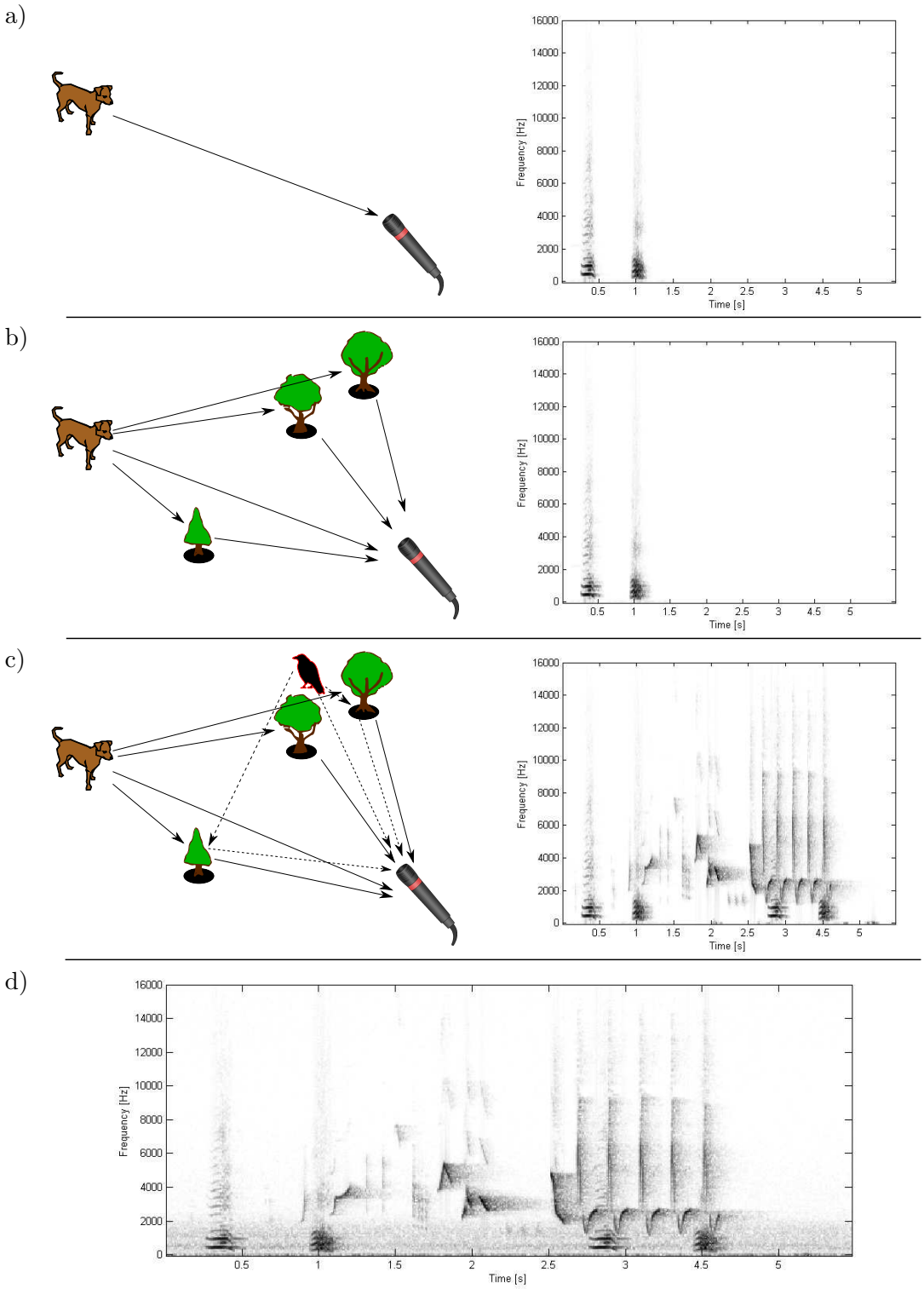


Figure 1.1: Important effects in the formation of complex audio scenes. a) Undisturbed signal: The barking of a dog recorded in an anechoic chamber. b) Multiple reflections of the signal lead to echo effects, *smearing out* the signal. c) The presence of multiple sound sources at the same time leads to an increase in complexity. In addition to the repeated barking of a dog, the song of a nightingale is present. d) Background noise is added.

a nightingale is present. The signal emitted by each of these sources is influenced by the acoustics of the environment.

Until now, we have only discussed sound sources leading to concentrated patterns in the spectrum. There are, however, numerous sound sources whose effects are spread over large portions of the spectrum. Usually, these effects are collectively denoted as noise. Typical sources of noise in natural audio scenes stem from wind moving the leaves of nearby trees or impacting the microphone. Traffic from motorways, trains or planes creates considerable noise even in sparsely populated areas. Figure 1.1d shows the impact of noise on our audio scene. The spectrogram shown in this figure gives a good impression of most of the effects found in complex audio scenes. In this example, we have restricted the number of sound sources and the resulting audio scene is thus only moderately complex.

Overview

Chapter 2. The algorithmic analysis of complex audio scenes aims at deriving information about audio sources of interest from an audio scene. For this task, it is desirable to have a large amount of annotated recordings of such scenes at hand. We are particularly interested in natural audio scenes, where the presence of animal voices plays a central role. An invaluable source for this kind of data is found in the Animal Sound Archive at the Humboldt University of Berlin. Interesting audio sources are so numerous and variable in natural audio scenes, that the administration of this information is an intricate task in itself. Until recently, the access to the data of the Animal Sound Archive has been comparatively cumbersome, with metadata only available locally at the archive. To improve upon this situation, we have enhanced the Animal Sound Archive by a practical web-based platform for cooperative research in bioacoustics. This platform is described in Chapter 2. It allows to easily access sound recordings and metadata via the web. Administrative tasks, such as controlling user accounts and file permissions, are supported by the platform, allowing individually controlled data access. Users can submit their own recordings online and mechanisms are provided for quality control processes to be applied before incorporation of submitted data into the main database. External data collections can be connected and searched using a central interface. In addition to data access, the platform provides mechanisms for online annotation of recordings. A plug-in concept allows to easily integrate signal processing applications into the system. We present two examples for this concept. First, we have implemented an automatic speech detection algorithm which is a valuable tool for browsing spoken annotations present in the recordings. Second, an algorithm for audible watermarking of recordings allows to create previews of the data which allow to evaluate the contents and quality of the data without giving uncontrolled access to the data.

Chapter 3. Although recordings in animal sound archives are usually very well annotated by metadata, it is almost impossible to manually annotate all background sounds made by animals. Thus, although a considerable amount of recordings of a species may be present in a database, they are very difficult to find. Complementary to classical text-based querying of such databases, algorithms capable of automatically finding sections of recordings similar to a query by example therefore provide a promising approach to content-based navigation. In Chapter 3, we present algorithms for feature extraction, as well as indexing and retrieval of animal sound recordings. Our feature extraction algorithm is adapted to the typical curve-like spectral features characteristic of many animal sounds. Starting from an inverted-list-based

method for multimedia retrieval, we suggest a ranking scheme which makes this retrieval method suitable for similarity search. The resulting similarity search method is integrated in the web interface to the Animal Sound Archive described in Chapter 2.

Chapter 4. In the last section, we have illustrated a number of effects leading to the complexity of natural audio scenes. The most severe problems for pattern recognition in complex audio scenes stem from the multitude of sources that may be present. Whenever sources overlap in time and frequency, pattern recognition becomes extremely difficult. In Chapter 4, we therefore investigate methods incorporating multiple microphones into the task of breaking down audio scenes into simpler components. Source separation methods exploit the fact, that the sounds of different sources emerge from different positions in space. This leads to differences in sound level and time-delay with which sounds are recorded at different microphones and these differences help in the separation of sound sources. Such methods often aim at recovering every source in a scene as exactly as possible. This is not a realistic goal for natural audio scenes. We rather try to extract lower complexity components from an audio scene. Such components are modeled as linear combinations of the source signals recorded by multiple microphones. In order to judge the complexity of such a component, we define a measure based on the spectral flatness of a signal. Using multiple hypotheses tracking, we find low complexity components by locally optimising this measure. In particular, this approach allows to track varying mixing parameters.

Chapter 5. Above, we have stated that one of the goals of audio scene analysis is to extract information about the audio sources present in audio scenes. The most prominent sources present in natural audio scenes are animals. A natural question to ask is if we can automatically deduce whether and when the vocalisations of a given animal species are present in a recording. One application of algorithms capable of doing so lies in monitoring the success of nature conservation efforts. In this context, animals are often used as indicators for the state of their habitats. Therefore, typical monitoring tasks comprise counting the number of animals of a certain species or checking for the presence of individuals of a species. Currently, these tasks are carried out by volunteers, leading to a number of problems including a loss of comparability of results when the volunteer responsible for an area changes, and bad coverage of species vocalising at night. Moreover, monitoring vocalising animals needs a thorough knowledge of animal voices. Finally, some biological reserves are not allowed to be entered in times when monitoring them is most interesting. In these situations, recording devices combined with pattern recognition algorithms could provide helpful supporting tools.

Chapter 5 deals with the problem of detecting the vocalisations of certain target species in audio recordings. Specialised algorithms for the detection of calls and songs of the Eurasian bittern, Savi's warbler, and the chaffinch are presented. The first two species are endangered inhabitants of reed bed areas. Monitoring their presence is helpful in assessing the health of such areas. The chaffinch is examined as an example of a bird with a very variable but highly structured song which can be recognised because of its structure. Principles used in the design of these special purpose algorithm are applicable to the recognition of other species showing similar call types.

Chapter 2

Web-based Cooperation in Bioacoustics Research

Aiming at analysing complex audio scenes, it is desirable to have at one's disposal a large amount of recordings of such scenes. As a field of application, we are particularly interested in natural audio scenes where the presence of animal voices plays a central role. An invaluable source for this kind of data is found in the Animal Sound Archive at the Humboldt University of Berlin. On the one hand, it comprises a large number of dedicated recordings of voices of a high number of target species. On the other hand its research interest in animal sound monitoring leads to a considerable number of long recordings of complex natural audio scenes. We have enhanced the Animal Sound Archive by a practical web-based platform for cooperative research in bioacoustics. Among other features, it allows for easy automated access to sound recordings for the purpose of audio scene analysis. In this chapter, we describe in some detail the goals and features of this platform.

2.1 The Animal Sound Archive

The Animal Sound Archive at the Humboldt University of Berlin [FBKC06] is one of the oldest and most extensive collections of animal sounds in the world. Founded in 1951 by Prof. Günter Tembrock, it now holds about 110,000 recordings of animal sounds on more than 4,500 magnetic tapes, DAT cassettes and CDs. The collection covers 1,800 species of birds, 580 species of mammals, more than 150 species of arthropods, as well as recordings of fish, amphibian and reptile species. In 2002, digitisation of the material has been started and is proceeding at increasing speed, making the archive a valuable data source for digital signal processing research. Once finished, the archive will comprise about 5,000 hours of digitised audio or about 10 terabytes of data. Each recording comes with textual annotation stored in a database. In this way, information on recorded species, recording conditions, geographical location and much more are available.

In addition to collecting a wide variety of vocalisations dedicated to target species, research at the Animal Sound Archive is focused towards the monitoring of natural audio scenes. Systematic multichannel recordings in areas significant for wildlife conservation have produced hundreds of gigabytes of recordings suitable for digital signal processing research. They are especially useful for studies in multichannel audio processing and audio pattern recognition (see Chapters 4 and 5).

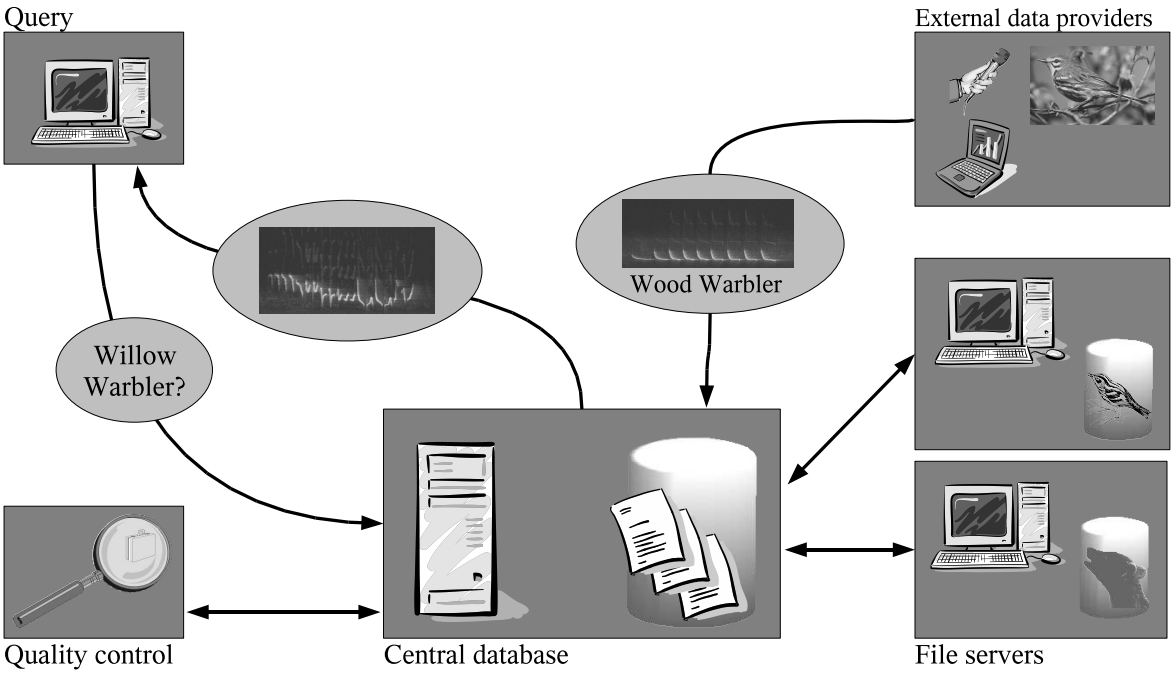


Figure 2.1: Overview of features for a cooperative research platform for bioacoustics.

2.2 Goals for a Cooperative Research Platform

Until recently, access to recordings of the Animal Sound Archive has been comparatively cumbersome. To obtain sound recordings, a request by mail — traditional or electronic — had to be addressed to the archive. At the archive, a search of the metadata database would reveal matching recordings which would then be copied to a CD and sent to the enquirer. This situation is not confined to the archive in Berlin. Worldwide, large animal sound archives are a rare case. Asking for electronic enquiry facilities and online access to audio data limits their number to virtually nil. To address these and other problems, we have enhanced the Animal Sound Archive by an information infrastructure supporting web-based collaborative research in bioacoustics in a project funded by the Deutsche Forschungsgemeinschaft (German Research Foundation) [BCFK05a, BCFK05b]. In this section, we will describe the goals of this endeavor.

Figure 2.1 gives a schematic overview of the desired features of a cooperative research platform for bioacoustics. The main question in designing a cooperative platform is where to store the data. There are two kinds of data involved, audio data and metadata, and the question has to be answered for both. Intensive consultation of practitioners in bioacoustics and potential users has led to the following decisions concerning the system's architecture: metadata for audio recordings is to be held in a central database together with a web platform for its access. Audio data on the other hand should remain with the respective creator or owner. Having a central storage of metadata makes enquiries to the metadata much easier to handle than otherwise. On the other hand, keeping audio data distributed leaves the provider with complete control over his data. It also reduces bandwidth load on file servers. However, it is not very convenient for providers of small amounts of data to maintain a file server solely for this purpose. Therefore, a mechanism for providers of small amounts of data is needed.

Data provided by such parties should undergo a quality control process before being included into the central database. Suitable means for that should be provided by the web platform.

A web interface should provide means for collaborative research on the data. Fundamental access to the data should be given by search via metadata such as animal species or geographic information. In addition, the interface should handle data delivery based on user specific access permissions. Additional services such as similarity search by audio content, time-stamp based online annotation of audio recordings and digital signal processing tools should turn the system from a mere archive to a platform for collaborative scientific research.

2.3 A Web-based Research Platform for Bioacoustics

We will now describe a web-based research platform for cooperative research in bioacoustics following the aims described in the previous section. Owing to the discriminate handling of audio data and metadata as described above, the system we have implemented is a combination of two modules, each dealing with one kind of data. First, we will describe the central web interface dealing with metadata and constituting the general user interface. Second, means for providing data for the infrastructure are detailed. Usually, providers of large amounts of audio data will do so by operating a file server. In the following, data provided in this way from one provider will be called a collection.

2.3.1 The Web Interface

User access to the information infrastructure is provided by a web interface. Password controlled user accounts are incorporated to manage individual access to metadata and audio files. A guest account[1] with limited permissions allows browsing and searching of metadata and access to a limited number of audio recordings. Active web pages have been implemented in a scripting language (PHP). All data — metadata, web page contents and user account data — is held in a relational database (MySQL). Services thus implemented comprise searching and browsing of metadata as well as audio data, providing and editing metadata, upload of audio data and tools for administrative tasks such as user management. We will now describe these services in more detail. Figure 2.2 gives an overview of the interface.

First, we exhibit the core functionality of the interface. The main purpose of the interface is giving access to metadata and audio data. There are three methods for metadata access. Firstly, traditional metadata-based search methods are supported. The most common search task, searching for all recordings of a given species exhibiting certain query terms in the description field of their metadata is offered as *standard search*. More detailed querying of metadata is offered as *extended search*. Here, a conjunctive query on a number of selectable database fields is possible. Secondly, metadata can be browsed by taxonomy. Finally, an interface for content based retrieval allows to upload a short audio recording which is used for similarity search, see Chapter 3.

All search methods should finally lead the user to metadata entries describing audio recordings of his interest. Details on such entries are provided in a dedicated view. Which part of the metadata a user will see and whether or not he will be able to edit information depends on the permissions assigned to the user's account. Figure 2.3 gives a comparison of this view

[1] The guest account to the research platform is accessible via http://www.tierstimmen.org

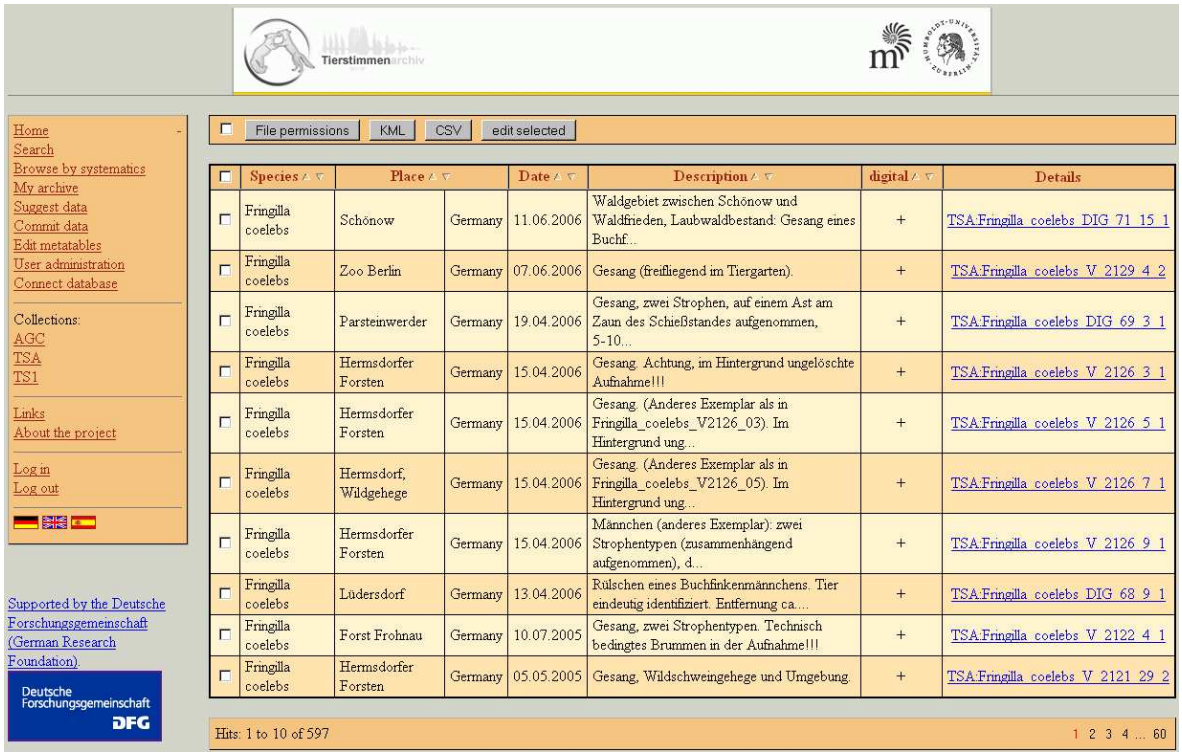


Figure 2.2: Overview of the web interface to the information infrastructure. The left part of the interface provides a menu comprising all functionality accessible by the user. These depend on the user's individual set of permissions. The right part shows a typical search result.

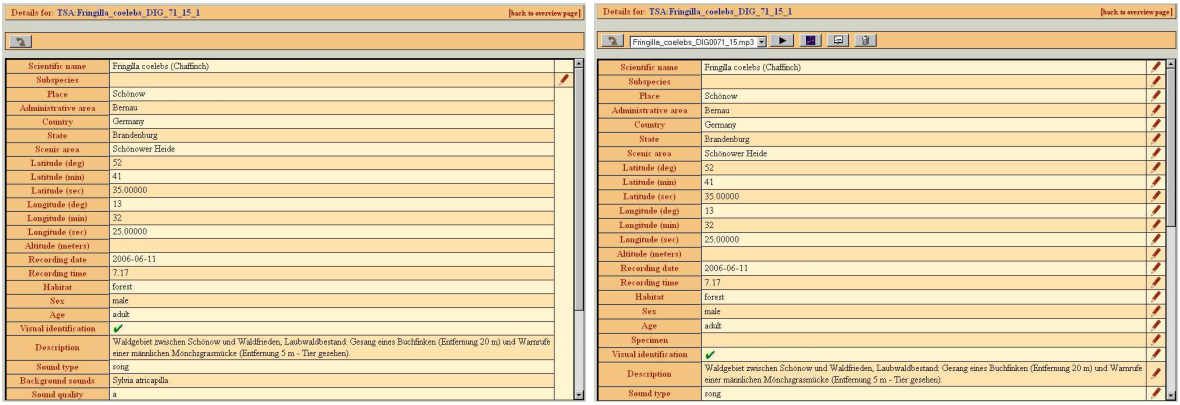


Figure 2.3: Metadata view for two users with different permission sets. The user on the left has restricted permissions, excluding the display of some of the metadata, editing capabilities, as well as downloading and browsing of audio data.

for users with different sets of permissions. From here, if permissions allow, the user is offered audio download. In addition to that, an applet allows online browsing of audio data using a spectrum view (Figure 2.4). The applet provides a number of additional features from audio playback to time based annotation of recordings and digital signal processing. Here, researchers can cooperate in annotating interesting features in audio recordings, species identification, etc.

Both, overview pages for search results and those for metadata can be extended by plugin tools. Plugins for the former typically handle lists of search results, whereas plugins for the latter work on single metadata entries. Plugins of the first group comprise tools for tasks such as administration of file permissions, export of result overviews as comma separated files or export of geographical coordinates for visualisation in Google Earth. Plugins for the second group include audio download and browsing as described above, reading and adding remarks for particular recordings and search for images of the current species.

A second functionality of the web interface is concerned with users wishing to provide small amounts of data to the archive. They usually do not want to maintain a webserver for this purpose. Therefore, a Java applet is supplied which allows to provide metadata entries to the central database and upload audio files to a collection provider's file server. One of the goals being reliable quality, no data provided in this way is incorporated into the database directly. It is first stored in a table for data suggestions and may then be explored and verified by particular users responsible for quality control.

Finally, the web interface provides a number of tools for administrative purposes. The most import administrative task is user management. The interface provides dialogs for the addition and removal of users, for modifying user permissions and handling of additional settings such as user accounts restricted to a fixed period of time. The infrastructure makes use of database tables detailing some of the entries of the main metadata table. As an example, information about the country in which a recording was made is stored as an ISO code in the main table. An additional table is used to translate these codes into the country's name in various languages. Data in these tables may be edited directly using the web interface.

In order to connect a file server providing a collection of audio data, some information has to be added to the infrastructure. These comprise a short identifier string for the collection,

the URL of the file server and flags describing which additional capabilities are supported. All information required for describing a connected collection can be administered via the web interface. Data providers are supported by a collection browser allowing to browse directories on their file server. It simplifies choosing files for setting permissions. Helpful information such as whether a metadata description is given for a submitted file is indicated by informative icons.

2.3.2 File Servers

Data providers hosting large amounts of audio data usually wish to maintain their own file servers. This allows for complete control of all provided audio data. A package of scripts allows file servers to be connected to the information infrastructure, thus providing audio data. In addition, new or updated metadata has to be transmitted to the central database. A spreadsheet-based tool is provided for this task allowing to import metadata from databases or comma separated files.

The main functionality provided by the file server scripts lies in the handling of audio files. Dedicated scripts deliver directory listings from file servers, stream audio data for data download and browsing, and manage uploaded files. All scripts authenticate user credentials and check user permissions prior to data delivery. Directory listings are generated to give the user a choice of different audio formats for download and for a file browsing tool supporting collection administrators.

For most of these tasks it is necessary for the file servers to communicate with the central database. A service script provides all the necessary functionality. First, it allows file server scripts to identify themselves to the central information system. Then, it provides methods for querying file permissions, check whether a user — identified by a session identifier — is administrator of a given collection, and enumerate user rights. Finally, it supports delivery of metadata from the central database.

In addition to the delivery of audio data, each file server can host a different set of additional tools operating on the data. The central database keeps track of such fileserver related information. Typically, such a tool is invoked as a plugin on one of the overview pages. Examples comprise tools for indexing and retrieval of audio data, for speech detection and audible watermarking (see Section 2.4). The tools for indexing and retrieval form the interface for the similarity search algorithms described in Chapter 3. The former invokes an indexing process to be run on the fileserver. The latter invokes a process performing an index-based search and writing search results to the database on the main server. Search results are displayed on a webpage which is updated regularly, when new data arrives in the database. The tools for speech detection automatically isolate spoken commentary in audio recordings and allow to listen to those parts of a recording only. This is helpful for checking the annotation of a file without having to listen to the complete recording. As a last example, we provide a tool for audible watermarking. It allows to give a first impression of the recordings on a file server to a large audience without giving away the recordings uncontrolled.

In order to simplify the development of fileserver tools, we provide a software development kit (SDK) for typical tasks. These comprise reading program parameters sent using the CGI standard, handling directory listings from the file system, reading audio data from files and parsing initialisation files. Together with a larger toolbox developed by the author, numerous tools for audio processing and analysis are available.

2.4 Signal Processing for Animal Sound Archives

In this section we describe applications of signal processing algorithms for annotating and delivering data from the Animal Sound Archive. First, automatic speech detection allows to easily find spoken comments in recordings. Second, audible watermarking allows to generate distorted previews of audio content in order to prohibit unauthorised reproduction.

2.4.1 Speech Detection

Recordings at the Animal Sound Archive usually contain comments spoken by the recordist. Although most recordings have spoken comments at the beginning or the end, comments are often found in other positions, too. It is very helpful if these comments can be found automatically such that they can either be easily accessed or skipped, depending on the task at hand.

Speech detection in general is not an easy task and much work has been devoted to this subject. It would however lead us too far astray to survey even some of the methods involved. For the Animal Sound Archive, our goal was to provide a simple yet reliable algorithm for speech detection in animal sound recordings. We have implemented such an algorithm which works by extracting feature vectors describing autocorrelation in a frequency band typical for human speech. These feature vectors are then classified using a Support Vector Machine (SVM). We will first describe the features used and then discuss the application of SVMs.

Let $S \in \mathbb{R}^{F \times T}$ contain the absolute values of the discrete windowed Fourier transform of an audio signal consisting of F frequency bins and T time positions. Features will be extracted from the frequency bins f in the range $f_\ell \leq f \leq f_u$. For each time position $t \leq T - \tau$, we examine a smoothed version of this frequency band over τ time positions: Let $\tilde{\sigma}(t) := \left(\sum_{i=0}^{\tau-1} S(f_u, t+i), \ldots, \sum_{i=0}^{\tau-1} S(f_\ell, t+i)\right)^\top$. Then, we regard $\sigma(t) := \tilde{\sigma}(t)/\|\tilde{\sigma}(t)\|$ as time smoothed vector for the given frequency band and extract features from the vectors $\sigma(t)$. We now compute a autocorrelation values of σ for each t as follows: For each $i \in \{1, 2, \ldots, a\}$ let $\alpha[\sigma(t)]_i := \sum_{f=i+1}^{a} \sigma(t)_f \sigma(t)_{f-i}$. In the following, $\alpha(t)$ will denote the vector $(\alpha[\sigma(t)]_1, \ldots, \alpha[\sigma(t)]_a) \in \mathbb{R}^a$. The vectors $\alpha(t)$ form the feature vectors of a given audio signal and we will use $\alpha(t)$ to decide whether speech is present at time step t.

For the discrimination of feature vectors $\alpha(t)$ into the classes *speech* and *non-speech* we use a Support Vector Machine [Vap98]. Support Vector machines are linear discriminant classifiers. Given a linearly separable training set of vectors, each assigned to one of two classes, a discrimination (hyper-)surface is found such that in each half-space defined by the surface only vectors of one class are present. Additionally, the distance from the surface of those training patterns closest to it is maximised. This leads to a quadratic programming problem where the vectors of the training set are used only in scalar products. The latter may be replaced by so-called kernel functions, thus introducing non-linearity. The quadratic programming problem to be solved is quite large in the sense that one Lagrange multiplier is introduced for each training vector. For SVM training we use the Sequential Minimal Optimisation algorithm [Pla99], an algorithm that solves this large quadratic programming problem by splitting it into smaller subproblems each considering only one pair of the Lagrange multipliers. It thus avoids keeping in memory the Hessian involved in this problem, which would take space quadratic in the number of training vectors.

Support vector classification leads to a collection of time intervals describing those portions of a recording that most probably contain speech. Some post-processing is needed to eliminate

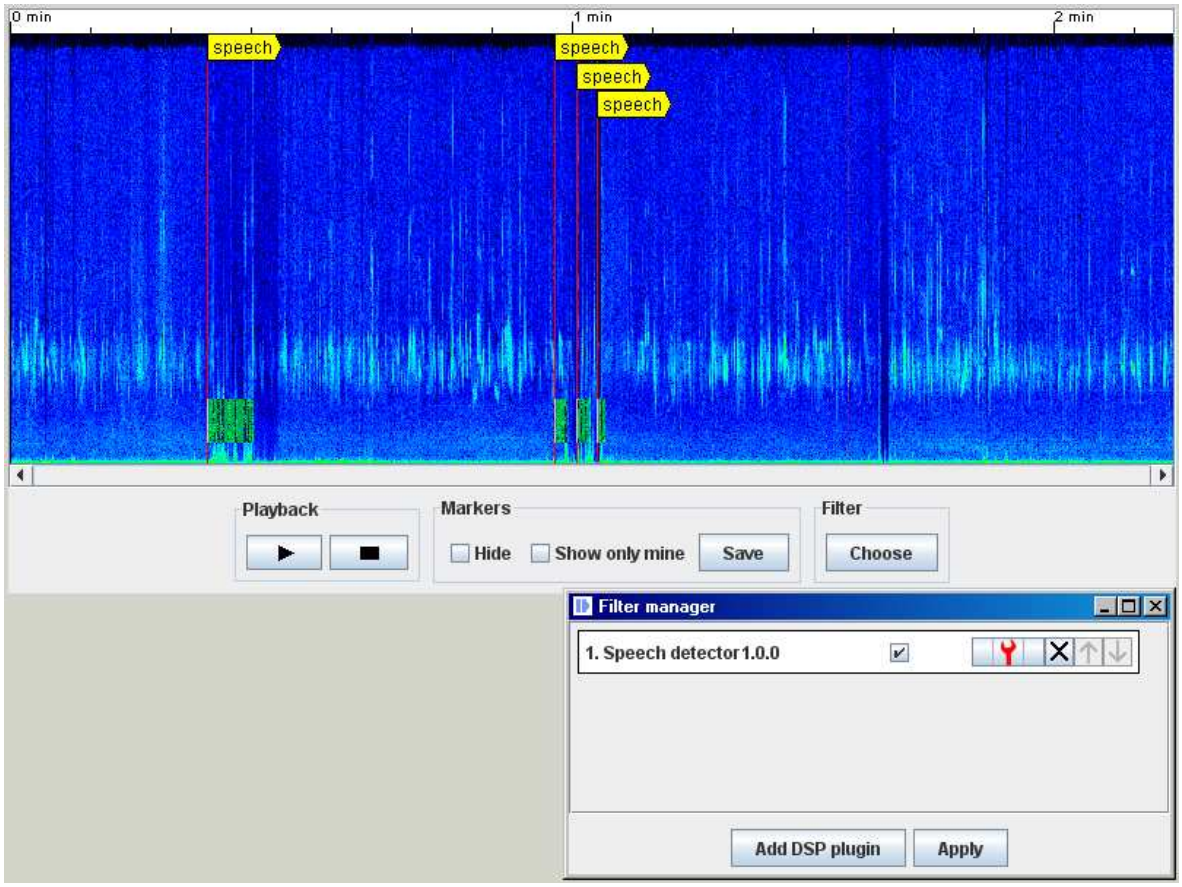


Figure 2.4: The speech detection algorithm as a plugin for the spectrum browser applet. Highlighting of detected speech makes skipping or deliberate choice of speech sections easy.

intervals that are too short as candidates for speech. Additionally, intervals in close proximity to each other will be merged and finally intervals will be slightly extended to ensure that beginnings and endings of speech utterances will be enclosed.

2.4.2 Audible Watermarking

In order to achieve an audible watermark that allows a sensible preview of audio contents but prevents unauthorised use of a recording or the image of its spectrum, we decided to embed readable text into the spectrum. This leads to a characteristic audible distortion of the audio file thus making clear to the listener that a watermark is present. Images of the spectrum reveal the embedded text.

Embedding a watermark works as follows. First, we perform the discrete windowed Fourier transform of a discrete signal. Then, we create a pixel image of the text to be embedded. The text will be repeated horizontally as well as vertically such that the pixel image has the same dimensions as the spectrum. To make removal more difficult, the vertical positions of the letters should be randomly displaced. Now, each spectrum entry will be replaced by a convex combination of the original spectrum and the image to be embedded. Some care has to be taken to create sensible phase information.

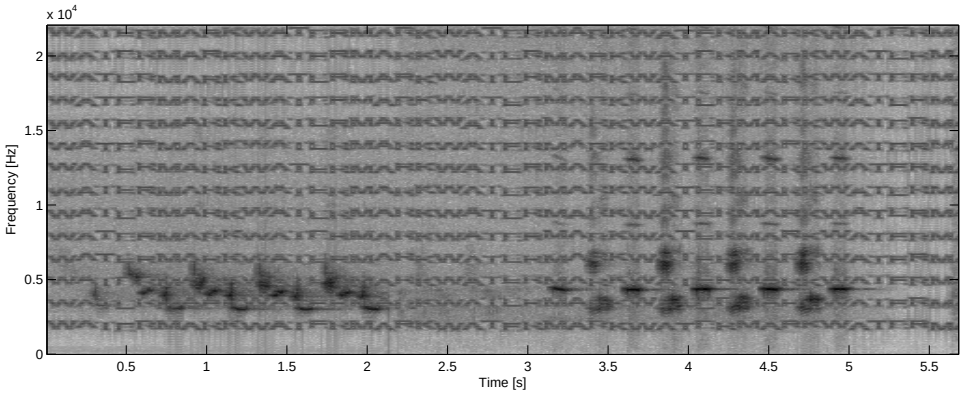


Figure 2.5: The spectrum of an audio recording with the text *WATERMARK* embedded.

To be more precise, let $S \in \mathbb{C}^{F \times T}$ be the discrete windowed Fourier transform of an audio signal we wish to furnish with a watermark. Let $I \in \mathbb{R}^{F \times W}$ be the image to be embedded. A parameter $\epsilon \in (0, 1)$ is chosen to steer the loudness of the embedded signal. $\epsilon = 0$ would not embed the image at all, $\epsilon = 1$ would replace the actual signal by the embedded image. The watermarked spectrum $\tilde{S} \in \mathbb{C}^{F \times T}$ is defined by the following formulas:

$$\begin{aligned} \mathrm{Re}(\tilde{S}_{f,t}) &= (1 - \epsilon) * \mathrm{Re}(S_{f,t}) + \epsilon * \cos(\theta(f)) * I_{f,t \bmod W} \\ \mathrm{Im}(\tilde{S}_{f,t}) &= (1 - \epsilon) * \mathrm{Im}(S_{f,t}) + \epsilon * \sin(\theta(f)) * I_{f,t \bmod W} \end{aligned}$$

Here, $\theta(f)$ is chosen such that the image magnitude $I(f, t \bmod W)$ represents a short sinusoidal tone of the form $I(f, t \bmod W)e^{\theta(f)}$ where $\theta(f)$ is the frequency associated to the f-th Fourier coefficient. Finally, the watermarked signal is obtained by the inverse windowed Fourier transform of the watermarked spectrum.

Figure 2.5 shows an example of the spectrum of an audio recording with an embedded text. Watermarked audio files sound like having some Morse code-like rhythm superimposed. Audibility and visibility of the watermark depend on the embedding strength ϵ. Using a suitable embedding strength it is still possible to judge contents and quality of a recording while the watermark is clearly audible and visible.

Chapter 3

Similarity Search in Animal Sound Databases

We are now in a position to have a large, easily accessible database of animal sounds at our disposal. We have already described means, manual and automatic, to enrich such a database with additional metadata. For example, we can automatically extract spoken comments. In this chapter we will examine similarity search in audio databases. By this, we mean that given an example recording of a sound, we would like to find all similar sounds in the database together with their exact time positions. This will again allow us to extract information inherent in the audio recordings but not easily found by manual methods.

3.1 Overview

Finding documents in a database that are in some way similar to a given query document is an important research question in many fields. After satisfying algorithms for text and web retrieval have already been available for some time [ZM06], non-standard documents like images, audio and video move more and more into the centre of attention.

Although different approaches are found in different domains like image search [SK01, LCL04, RMV07, DJLW08], audio matching [WBKW96, Cas01, AHFC01, CBMN02, GL03, KM08], video retrieval [VBK01, LOSX06, ZZS07] and search in 3d data [FMK$^+$03, CTSO03, NK04, Mos01], there are some typical steps central to most such approaches. We will shortly describe an approach by Mitra et al. [MGGP06] to similarity search in 3d data which is well suited for illustrating these core steps.

Mitra et al. address the problem of similarity search in collections of shapes, i.e., 2-dimensional submanifolds of $\mathbb{R}^3$. They describe a process of extracting fingerprints of such shapes which can then be compared to find similar shapes. These fingerprints are found by the following steps. First, a given shape is uniformly sampled by a fixed number of points. Then, it is segmented into so called shingles by placing a sphere of fixed radius at each sample point and defining the corresponding shingle to be the intersection of this sphere with the shape. After this central step of segmentation, local features are extracted from each shingle by computing spin-images — features derived from the relative position of points on a shingle to the surface normal at the centre of the sphere defining the shingle. Thus, a shape is represented by the point positions and the corresponding spin images. Finally, more compact fingerprints are obtained by a hashing scheme. This procedure exhibits three core steps for

similarity search systems: segmentation of the original objects, extraction of local features from each segment and incorporation of segment information and local features into compact database objects.

Our approach to similarity search, which is composed of similar steps, stems from a general group theoretic method for content-based multimedia retrieval. We will now give a short review of this method. More detailed expositions can be found in [CKK03] and [Bar03].

A typical task to be addressed in multimedia retrieval can be formulated as follows. Let M be a set of elementary objects from which multimedia documents can be composed, i.e., multimedia documents are modeled as finite subsets of M. Moreover, documents can be modified to obtain similar documents — think of rotating a three dimensional structure or time-shifting an audio signal, here. All possible modifications conserving similarity are modeled by a group G together with an operation of G on M. This means, we have a group homomorphism Φ from G to the symmetric group on M. In particular, for each $g \in G$ and $m \in M$, we know the image $\Phi(g)(m) =: gm \in M$ of m under the permutation given by the homomorphism applied to g. Now let $\mathcal{D} := (D_1, \ldots, D_n)$ be a database of multimedia files $D_i \subset M$. Given a query-by-example $Q \subset M$, we wish to find all the ways in which we can manipulate Q by elements g of G such that $gQ := \{gq \mid q \in Q\}$ is a subset of one of the documents D_i. More formally, we are interested in computing the set $G_{\mathcal{D}}(Q) := \{(g, i) \mid gQ \subset D_i\}$. This can be done efficiently by using inverted lists as follows. First, Q can be evaluated element-wise, i.e.,

$$G_{\mathcal{D}}(Q) = \bigcap_{q \in Q} G_{\mathcal{D}}(\{q\}). \tag{3.1}$$

Now, note that M decomposes into disjoint equivalence classes according to the equivalence relation on M given by $m \sim m' \Leftrightarrow \exists g \in G : gm = m'$. Thus, each element $m \in M$ can be represented in the form $g_m r_m$ where r_m is a representative of the equivalence class of m and $g_m \in G$. Having fixed a system of representatives, the inverted list $G_{\mathcal{D}}(m)$ of an element $m \in M$ can be obtained from the list of r_m as

$$G_{\mathcal{D}}(\{m\}) = \{(gg_m^{-1}, i) \mid gr_m \in D_i\} =: G_{\mathcal{D}}(r_m)g_m^{-1}.$$

Putting this into equation (3.1) yields the results to a query Q as the intersection

$$G_{\mathcal{D}}(Q) = \bigcap_{q \in Q} G_{\mathcal{D}}(r_q)g_q^{-1}.$$

Our approach to similarity search in animal sound databases is derived from this framework. We will however replace the strict process of intersecting inverted lists by a more flexible ranking process.

Approaching similarity search in this way, there is no explicit definition of similarity. Rather, we choose a very pragmatic way around the problem of defining similarity: we describe documents by features whose numerical similarity in turn is used as the definition of similarity of the documents.

3.2 Feature Extraction

We will start our discussion of similarity search by describing the features to be extracted from each audio file and which will form the search index. Let $\mathcal{D} = (D_1, \ldots, D_n)$ be a collection

of audio signals where each $D_i \in \mathbb{R}^{\mathbb{Z}}$. From the spectrum of each signal D_i we extract a set of tuples $(i, f, t, c) \in \mathbb{Z}^4$ where c denotes a feature class at time position t in frequency band f. For this purpose, we first compute the discrete windowed Fourier transform of D_i. Then, we decide at which points of the spectrum a feature should be extracted, see Section 3.2.1. A feature class is then attributed to the neighbourhood of each such point. The feature class c is a 12-bit value describing the shape of the 2-dimensional Fourier transform of a given neighbourhood. More details on feature classes are given in Section 3.2.2.

3.2.1 Selecting Points of Interest

Most audio signals produced by animals (especially by birds) exhibit curve-like spectral components. We adopt a technique from image processing, the structure tensor [FG87], in order to identify portions of the spectrum that show this kind of structure.

Interpreting greyscale images as differentiable functions $I : \mathbb{R}^2 \to \mathbb{R}$, $(x, y) \mapsto I(x, y)$ we can define the partial derivatives $I_x(x, y) := \frac{\partial I}{\partial x}(x, y)$ and $I_y(x, y) := \frac{\partial I}{\partial y}(x, y)$. Then, the structure tensor is defined as

$$J_0(I)(x, y) := \begin{pmatrix} I_x^2(x, y) & I_x(x, y)I_y(x, y) \\ I_x(x, y)I_y(x, y) & I_y^2(x, y) \end{pmatrix}.$$

The structure tensor contains information about the dominant orientation of the image at a given pixel. It is given by the eigenvector of $J_0(I)(x, y)$ belonging to the largest eigenvalue. This information, however, is easily distorted by noise because it stems only from the direct neighbourhood of one pixel. In order to get more robust information, we use the mean of the structure tensors of pixels from a 11-by-11 pixel neighbourhood of a pixel.

Now we interpret the spectrum of an audio signal as an image and find those points where a strong dominant orientation is present. For this purpose, we compute the windowed Fourier transform of each input signal, using a Hann window of 32 milliseconds length with 50% overlap. Points of interest are computed using the logarithm of the absolute values of the transform. We divide this spectrum into frequency bands. Only the neighbourhoods of pixels at the center of each band are considered as possible points of interest. The width of the frequency bands is chosen such that the frequency width of the pixel neighbourhoods exactly covers the band.

We will decide whether a dominant orientation is present by calculating the largest eigenvalue of the smoothed structure tensor. The mean of a set of structure tensors is a symmetric matrix

$$M := \begin{pmatrix} A & B \\ B & C \end{pmatrix} \in \mathbb{R}^{2\times 2}$$

and its eigenvalues are the zeros of its characteristic polynomial $\chi_M(\lambda) = \lambda^2 - (A + C)\lambda + AC - B^2$. We now assign an *attention value* $a(x, y)$ to a given pixel if M has two distinct eigenvalues, i.e. if the discriminant of χ_M, $D := \frac{(A+C)^2}{4} - AC + B^2$, is larger than zero. Then let $a(x, y) := \frac{A+C}{2} + \sqrt{D}$, otherwise let $a(x, y) := 0$.

The final decision whether a feature is extracted at a point (x, y) is based on a local threshold θ_1, i.e., a feature is extracted at each point with $a(x, y) > \theta_1$. In order to cope with audio signals that are too large to be kept in memory, the analysis of the audio signals is performed in a windowed manner, analysing segments of 1000 spectrogram columns and a small overlap corresponding to the width of analysis windows used for the 2d Fourier

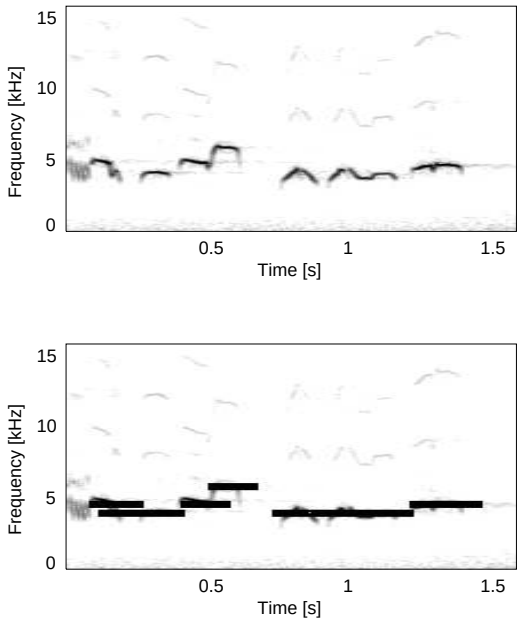


Figure 3.1: Points of interest (bottom) selected from an excerpt of the spectrum of Turdus merula (top).

transform. A new threshold is computed for each such segment. The threshold is found by regarding a histogram h of a counting the number of points in a window with values of a in a given range. More precisely, we divide the interval $[0, \max_{x,y} a(x,y)]$ into 1000 bins and define the histogram h as

$$h(\ell) := \sharp\{(x,y) \mid \ell \leq 1000 \cdot a(x,y)/\max_{x,y} a(x,y) \leq \ell + 1\},$$

$\ell \in \{0, 1, \ldots, 999\}$. Now, we choose ℓ such that $\sum_{i=1}^{\ell} h(1000-i) \geq p \sum_{i=0}^{999} h(i)$. From this, we obtain θ_1 by scaling ℓ to the range of a: $\theta_1 = \frac{\ell}{1000} \cdot \max_{x,y} a(x,y)$. A practical value for p is $p = 0.96$.

Figure 3.1 shows points of interest selected from the spectrum of a Blackbird song.

3.2.2 Feature Classes

We now proceed to assigning a feature $(i, f, t, c) \in \mathbb{Z}^4$ to each position (f, t) in the spectrum of the signal D_i determined by the procedure described in the previous section. The frequency-band f and the time-position t are determined by the position in the spectrum. In the following, we will describe how the feature class c is derived.

The feature class c is a 12-bit value, each bit describing an aspect of the geometric shape of the 2-dimensional Fourier transform of a 16-by-16 pixel neighbourhood of a point of interest. The feature class of a point p is obtained as follows (see Figure 3.2). First, compute the discrete 2-dimensional Fourier transform of the 16-by-16 pixel neighbourhood of the point, containing the point p at the upper left pixel of its central four pixels (see Figure 3.2a and d). This neighbourhood is divided into four columns, each 4 pixels wide (Figure 3.2b and e). The four subcolumns of each column are concatenated to four vectors $v_1, \ldots, v_4$ of length 64.

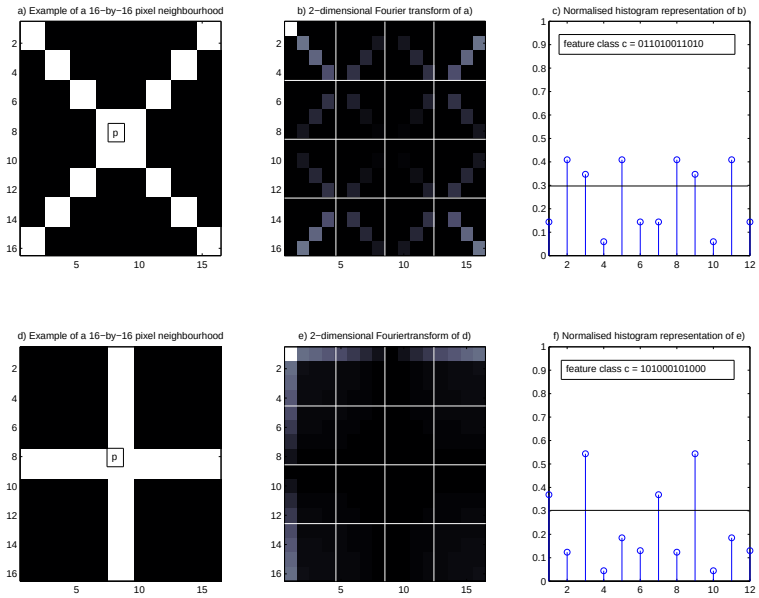


Figure 3.2: Deriving a feature class from the neighbourhood of a pixel p. Parts a) and d) give examples of such neighbourhoods from Figure 3.1. The respective 2d Fourier transforms are given in parts b) and e). These are divided into 4-by-16 element rows and columns whose pairwise dot products constitute the histogram representations given in parts c) and f). The feature classes in the latter parts are found by assigning a bit 1 or 0 to each histogram value depending on whether it is larger than the threshold (horizontal line) or not.

The six possible pairwise scalar products $v_i^\top v_j$ $(i \neq j)$ of the v_i comprise entries one to six of a new vector $s \in \mathbb{R}^{12}$. Entries seven to twelve are obtained similarly, considering rows instead of columns. Unfortunately, the representation of images by the 2d Fourier transform is quite unintuitive and hence it is hard to read off the characteristics of an image of its transform.

Finally, the feature class c is obtained by first normalising s with respect to the Euclidean norm and then setting bit i of c to 1 if $s_i > 0.4$. All other bits are set to zero (Figure 3.2c and f[1]).

Using the shape of the Fourier transform to define the feature class gives some robustness towards small changes of the pixel position from which the feature class is derived. Reducing the information from the Fourier transform to a bit-feature allows us to define the distance between two feature classes as the number of differing bits. This is very easy to compute and allows fast retrieval of all features similar to a given feature class from a database.

3.3 Indexing and Retrieval

For similarity search, we use an indexing scheme derived from the general principle sketched in Section 3.1. The set $M := \{1, \ldots, F\} \times \mathbb{Z} \times \{0, \ldots, 2^{12} - 1\}$ comprises the basic elements

[1] A threshold of 0.3 instead of 0.4 is used in the figure because it gives a better illustration of the thresholding process.

for document description. The first component describes F frequency bands, the second component takes track of time and the final component indicates feature classes. The group $G = \mathbb{Z}$ serves to model time shifting, allowing to find the position of a query in a recording. It operates by addition on the time component of M. In this way, our modelling of similarity search is completely within the feature domain.

With feature extraction producing index objects (i, f, t, c), indexing simply amounts to storing all the index objects obtained from a document collection in a relational database. From this, we will be able to restore inverted lists $\mathcal{I}(i, f, c)$ to each triple (i, f, c) consisting of all time positions at which feature class c was detected in frequency band f of document D_i. Storing inverted lists for each document separately allows us to compute a ranking for one document at a time.

Following the framework introduced in Section 3.1, in order to answer a query $Q = \{(f_1, t_1, c_1), \ldots, (f_q, t_q, c_q)\}$ we would retrieve, for each document D_i, the inverted lists for each query element $(f, t, c) \in Q$, shift the corresponding time values by t and find the positions where the query matches the document by intersecting these adjusted lists.

We use two mechanisms to make this scheme suited for similarity search instead of exact matching. First, for a triple $(f, t, c) \in Q$, we merge all inverted lists $\mathcal{I}(i, f, c')$ having a feature class c' differing from c in at most Δ bits, where Δ is a parameter adjustable by the user. Second, we replace exact list intersection by a ranking scheme as follows. For each document D_i, we will compute a ranking function $r_i : \mathbb{Z} \to \mathbb{Z}$. For time position t in document D_i, $r_i(t)$ describes how similar the query is to the document at that position.

For each time position t in one of the inverted lists fetched for a query element, we add a local ranking function $r : \mathbb{Z} \to \mathbb{Z}$ with local support to r_i centered at position t. More precisely, given the inverted lists $\ell_{i1}, \ldots, \ell_{im_i}$ for document D_i, we compute the ranking function r_i as follows:

$$r_i = \sum_{j=1}^{m_i} \sum_{t \in \ell_{ij}} r(\cdot - t).$$

An example for computing the ranking function of a document is given in Figure 3.3.

Finally, a small number of positions with high ranking values should be reported to the user. These positions are chosen in two steps.

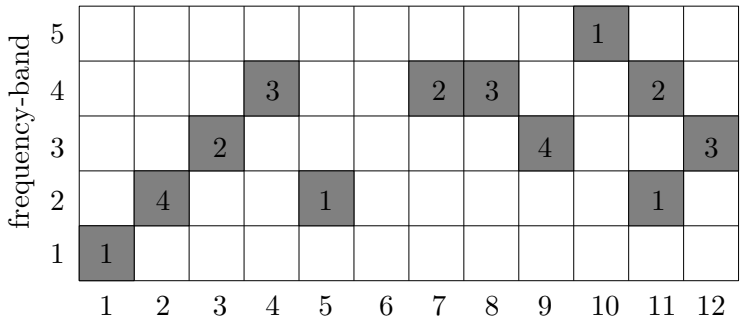
First, peak picking is performed on the ranking function after a low pass filter has been applied. For each position t, we decide whether a peak is found at that position by regarding the ranking function at positions $t-k, \ldots, t-1$ on the one hand and the positions $t+1, \ldots, t+k$ on the other hand. The slope $m_\pm$ of the regression line minimising $\sum_{i=1}^{k} (m_\pm i + b - r(t \pm i))^2$ is found for both parts and a peak is found whenever the slope m_- for the left part is positive and the slope m_+ for the right part negative. The slope of the regression line is found by solving the respective normal equations. For our optimisation problem, they are given by

$$A^\top A \begin{pmatrix} m_\pm \\ b \end{pmatrix} = A^\top \begin{pmatrix} r(t \pm 1) \\ \vdots \\ r(t \pm k) \end{pmatrix} \text{ for } A = \begin{pmatrix} 1 & 1 \\ \vdots & \vdots \\ k & 1 \end{pmatrix}.$$

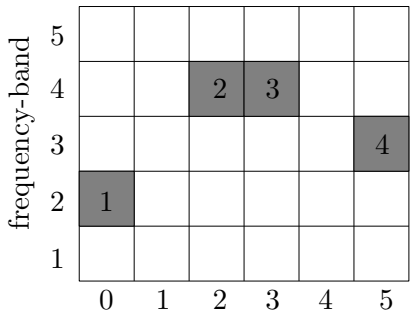
This leads to the following system of linear equations

$$\begin{pmatrix} \sum_{i=1}^{k} i^2 & \sum_{i=1}^{k} i \\ \sum_{i=1}^{k} i & k \end{pmatrix} \begin{pmatrix} m_\pm \\ b \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{k} i r(t \pm i) \\ \sum_{i=1}^{k} r(t \pm i) \end{pmatrix},$$

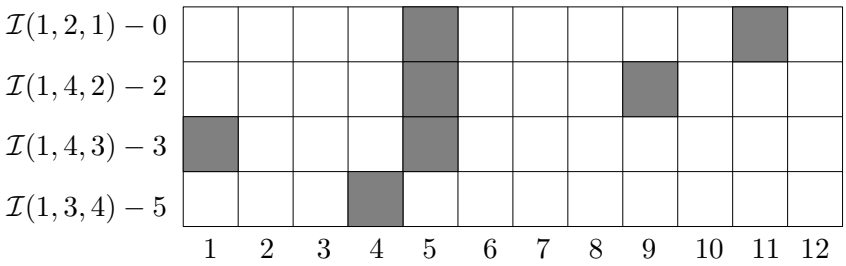
Features extracted from **document** D_1.



Features extracted from a **query**



Inverted lists



Ranking function $r_1(t)$

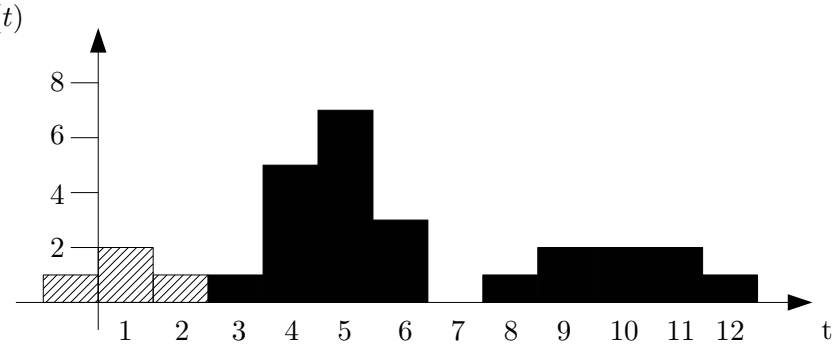


Figure 3.3: Computing the ranking function of a document. $\mathcal{I}(i, f, c) - t$ denotes the inverted list for file i, frequency-band f and feature class c with t subtracted from each entry of the list. The local ranking function r, translated to be centered at 1, is indicated as the dashed part of the graph of r_1. It is defined by $r(-1) = 1$, $r(0) = 2$, $r(1) = 1$, and $r(x) = 0$ otherwise.

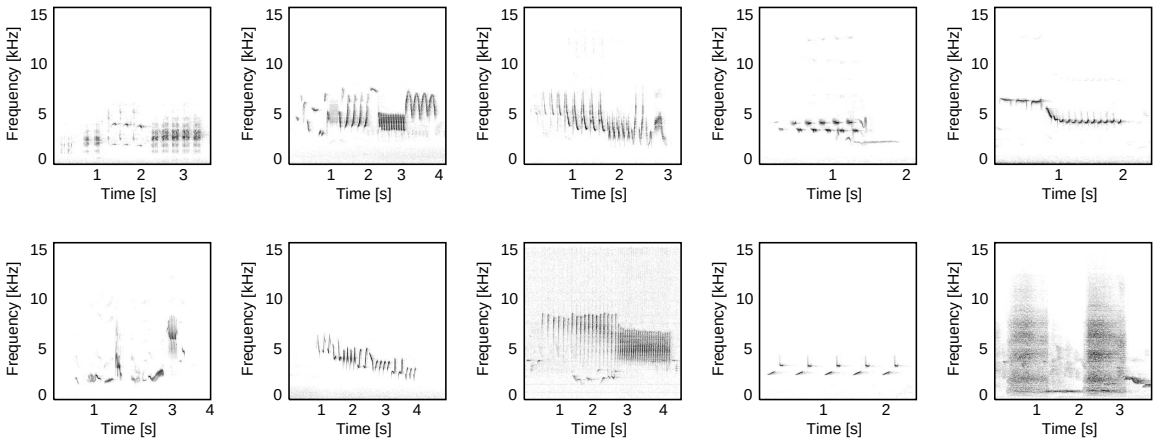


Figure 3.4: Ten of the test queries showing the spectral characteristics of the different bird vocalisations used in the retrieval tests. Upper row: Acrocephalus arundinaceus (Great Reed Warbler), Troglodytes troglodytes (Wren), Fringilla coelebs (Chaffinch), Lullula arborea (Wood Lark), Parus cearuleus (Blue Tit). Lower row: Turdus merula (Blackbird), Phylloscopus trochilus (Willow Warbler), Phylloscopus sibilatrix (Wood Warbler), Parus major (Great Tit), Ailuroedus buccoides (White-eared Catbird)

which, when solved for $m_{\pm}$, leads to

$$m_{\pm} = \frac{\sum_{i=1}^{k} ir(t \pm i) - (\sum_{i=1}^{k} r(t \pm i))^2}{\frac{k}{6}(k+1)(2k+1) - k(\frac{k+1}{2})^2}.$$

After peak picking, as a second step, the ten highest ranking peaks from each document are chosen, subject to the condition that no two peaks are closer together than the query length.

3.4 Results

We have tested our search algorithm on an index created from 1000 manually selected files from the Animal Sound Archive of the Humboldt University, Berlin. The indexed test dataset comprises 20 hours of animal recordings from 264 species. The time required for indexing a file depends very much on the number of interest points selected. Usually, indexing takes a quarter of the playback time of a recording. The index is stored in a MySQL database and uses 260MB of space for 1.6GB of MPEG 1 Layer 3 compressed audio files.

It is very laborious to obtain ground truth data for data sets of the given size. A manual annotation of such data sets requires a high amount of expertise in the recognition of animal sounds as well as an extraordinary amount of time because all of the recordings have to be played back and processed in real time. We therefore tried to use for evaluation as much of the information given by the metadata as possible. Most importantly, each recording is labeled with an animal species which was the target for the recording. Additionally, many recordings come with a list of species vocalising in the background.

For a first test, we chose ten species of bird and tested whether species identification by similarity search is possible. More specifically, we chose five songs from each bird and

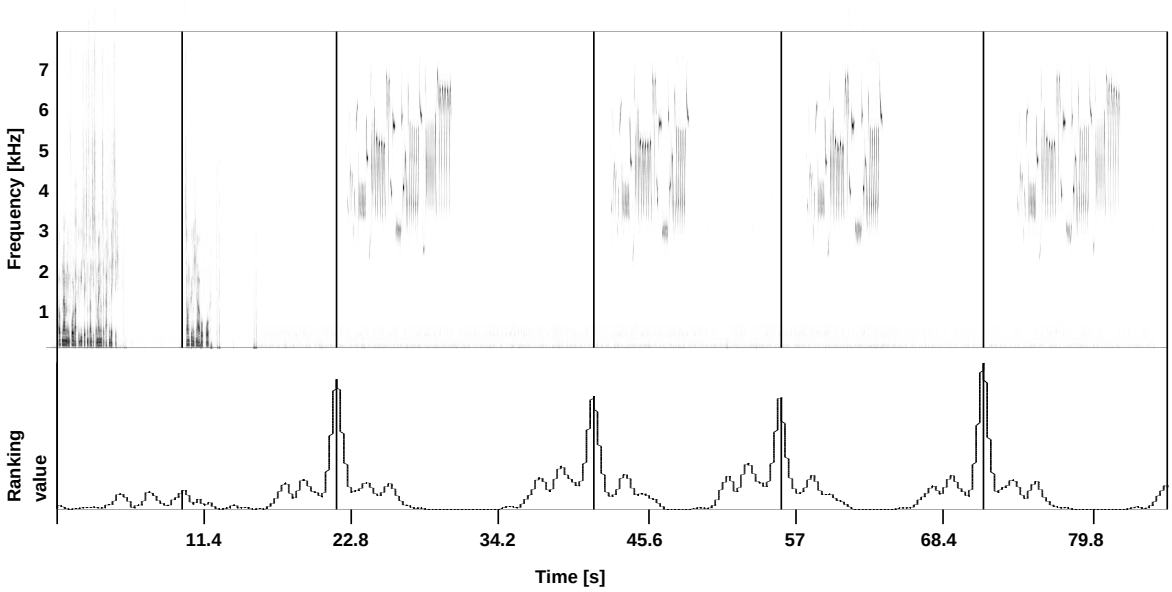


Figure 3.5: An example for a retrieval result for the song of Troglodytes troglodytes. Spectrogram (top) of the first 81 seconds of a recording containing variations of this song. Vertical lines mark positions corresponding to peaks in the ranking function (bottom) selected by the algorithm.

tested whether the correct species is found among the top ranks of the retrieval results. The test recordings where not contained in the recordings used to create the index. Figure 3.4 shows the spectral characteristics of the vocalisations of the ten species used for evaluation. Table 3.1 gives a summary of the results. For each query, we have given the first position in the ranking list that belongs to a recording labeled with the species from which the query was taken. Moreover, we give the relative ranking value of this hit, i.e., the quotient of the hit's ranking value and the ranking value of the highest ranking result. The parameter $\Delta = 2$ was chosen for these tests.

The quality of the results depends on the kind of vocalisations used as a query. By design, the feature extraction process works best for signals with curve-like spectral characteristics. This is reflected by the high ranking results for queries of vocalisations of Fringilla coelebs, Phylloscopus trochilus, Phylloscopus sibilatrix, and to some extent Lullula arborea and Troglodytes troglodytes. In these cases, false positives in the first ranks usually stem from structurally similar songs. For example, the songs of Phylloscopus sibilatrix and Fringilla coelebs are very similar and therefore lead to high ranking results for both species. Figure 3.5 shows the ranking curve and the selected result positions for a query of a Troglodytes troglodytes song.

Ailuroedus buccoides was chosen for its noise-like spectral characteristics (see Figure 3.4, upper left spectrogram) which is not very well suited for the proposed features and is therefore well suited for finding out the limits of our method. Nevertheless, retrieval results are quite good in this case. High ranking false positives are less obviously related to queries than in the previous case. They do however tend to show either similar repetition patterns to the query or noise like structures with formant-like maxima.

query number	species	query length	highest rank of same species	relative rank
1	Acrocephalus arundinaceus (Great Reed Warbler)	3.7s	58	0.616
2	”	5.4s	70	0.61
3	”	7.6s	99	0.6
4	”	6.5s	43	0.68
5	”	4.7s	135	0.54
6	Troglodytes troglodytes (Wren)	4.2s	2	0.95
7	”	7.1s	83	0.51
8	”	16.4s	4	0.75
9	”	6.9s	3	0.85
10	”	6.5s	10	0.61
11	Fringilla coelebs (Chaffinch)	3.3s	10	0.74
12	”	3.6s	1	1
13	”	4.5s	2	0.97
14	”	3.3s	1	1
15	”	3.3s	1	1
16	Lullula arborea (Wood Lark)	2.2s	36	0.57
17	”	3.8s	1	1
18	”	4.0s	4	0.96
19	”	4.0s	7	0.85
20	”	4.6s	2	0.98
21	Parus cearuleus (Blue Tit)	2.5s	10	0.75
22	”	2.8s	15	0.76
23	”	5.1s	18	0.63
24	”	2.2s	1	1
25	”	3.5s	83	0.51
26	Turdus merula (Blackbird)	3.5s	41	0.64
27	”	3.8s	17	0.67
28	”	3.7s	43	0.71
29	”	4.0s	11	0.87
30	”	4.5s	41	0.6
31	Phylloscopus trochilus (Willow Warbler)	5.0s	1	1
32	”	4.4s	1	1
33	”	5.3s	1	1
34	”	7.3s	1	1
35	”	18.0s	1	1
36	Phylloscopus sibilatrix (Wood Warbler)	3.9s	7	0.87
37	”	4.6s	1	1
38	”	6.3s	1	1
39	”	7.6s	2	0.79
40	”	5.8s	3	0.87
41	Parus major (Great Tit)	11.5s	504	0.49
42	”	4.8s	334	0.45
43	”	4.2s	376	0.34
44	”	5.1s	104	0.66
45	”	3.2s	263	0.53
46	Ailuroedus buccoides (White-eared Catbird)	5.4s	18	0.75
47	”	2.9s	27	0.68
48	”	3.8s	4	0.86
49	”	7.5s	11	0.7
50	”	5.5s	11	0.71

Table 3.1: Retrieval results for five songs of each of ten species of birds. For each query recording, we give its length, the highest rank of a recording of the same species within the retrieval results and the quotient of the ranking value of each hit with respect to the highest ranking value for the query.

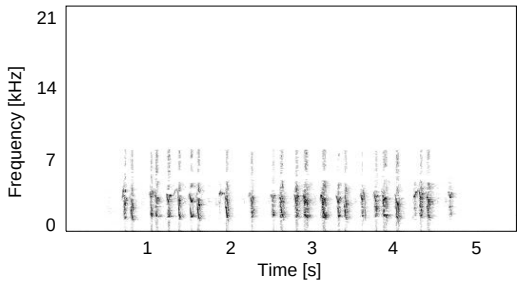


Figure 3.6: Spectrogram of the song of Falco eleonorae (Eleonora's Falcon). Queries of the song of Acrocephalus arundinaceus (Figure 3.4, upper left spectrogram) often lead to the retrieval of this kind of song.

Acrocephalus arundinaceus also shows noise-like features (Figure 3.4, lower right spectrogram). Moreover, it has more variation in the overall structure of its songs and thus leads to worse results than Ailuroedus buccoides. It is often confused with vocalisations produced by different species such as Falco eleonores (see Figure 3.6) containing repeated short noise-like elements. It also often leads to the retrieval of recordings with a strong noise background.

Turdus merula, showing spectral features that seem much better suited for our kind of features (Figure 3.4, lower left spectrogram), leads to less favourable results. This is explained by the high variability of the blackbirds's song. As queries are required to be of a certain length in order to be discriminating enough to give reasonable results, queries for blackbird songs comprise multiple syllables of the song. Because of the high number of syllables of the blackbird and the great number of ways in which these are combined, it is improbable to find a close match of any song in the database. On the other hand, it is easily confused with birds showing similar syllables and this can also be found in the results.

Close investigation of the results has been carried out in order to explain the bad retrieval results for Parus major. By examining the spectra of the retrieval results, it became clear that high ranking results usually belong to signal degradations resulting from signal clipping or starting and stopping of analogue tape recorders. These lead to sharp vertical lines in the spectrum which are detected as *curve-like features* by the feature extraction process. The vocalisations of Parus major are comprised of alternating higher and lower pitched notes. The higher pitched notes usually start with a sharp vertical line in the spectrum and thus match the signal degradations leading to similar effects.

Chapter 4

Array Processing and Source Separation

Most of the complexity of natural audio scenes stems from the multitude of source signals that comprise the scene. Thus, the first idea in dealing with complex audio scenes is to try to break them down into less complex parts. There are various ideas how to accomplish this and we will concentrate on methods incorporating multiple microphones into this task.

Initially, in this chapter we will review two approaches to the analysis of sound mixtures. First, beamforming exploits the spatial structure of an acoustic scene by using an array of microphones to create steerable directivity. Second, independent component analysis tries to invert linear mixtures of sounds by assuming that physically different sound sources should produce statistically independent signals.

Following this, we develop a method for source separation that is aimed to overcome some of the difficulties that arise in the application of source separation techniques to complex audio scenes. Its main features are local measures for separation success and the tracking of multiple separation hypotheses.

4.1 Beamforming and Source Localisation

Beamforming [VB88] is a method for separating sources that overlap in the frequency domain but come from different spatial locations. Using multiple omnidirectional microphones, it is possible to simulate the response of a directional microphone from the weighted combination of the responses of the undirected microphones. In this way, it is possible to attenuate those signals originating from directions different than that of a given source of interest.

Suppose, that we are recording an audio scene with a linear array of microphones with constant spacing (see Figure 4.1). There are two cues available which can help to concentrate on signals emanating from one direction. First, if the distance of the microphones is large compared to the propagation speed of sound, a source signal arrives at different microphones at different times. Second, sound is attenuated during propagation and thus a source signal arrives with different sound levels at different microphones. Both effects can be used to amplify signals from a given direction by constructive inference and attenuate signals from other directions by destructive inference.

Beamforming amounts to a combination of temporal and spatial filtering. First, the signal of each sensor is filtered in the time domain. Then, the resulting signals are linearly combined.

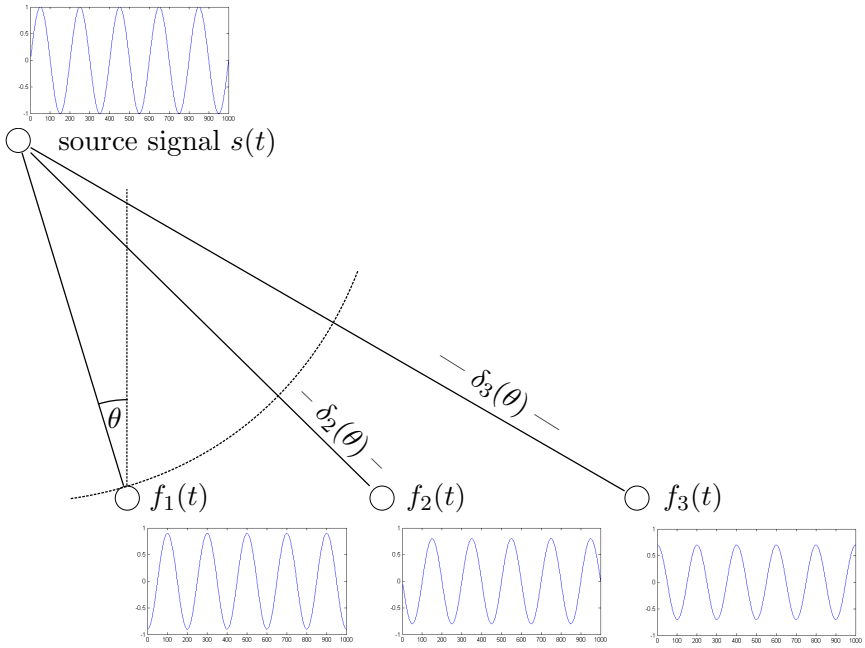


Figure 4.1: A source signal $s(t)$ received by a microphone array (f_1, f_2, f_3). Each sensor of the array receives the signal with a different time-delay and different attenuation. The time-delays depend on the differences $\delta_i(\theta)$ in the distance to the source, hence on the source bearing θ.

Hence, the beamformer output $b(t)$ of a microphone array of N elements, recording signals $f_1, \ldots, f_N$, is defined as

$$b(t) = \sum_{m=1}^{N} \alpha_m \sum_{k=0}^{L-1} c_{mk} f_m(t-k).$$

All filters are assumed to be of length L. Incorporating the signal weights α_m into the filter coefficients c_{mk}, i.e., defining new filter coefficients $w_{mk} := \alpha_m c_{mk}$, the beamformer output is expressed as:

$$b(t) = \sum_{m=1}^{N} \sum_{k=0}^{L-1} w_{mk} f_m(t-k).$$

Filtering in the time domain allows to account for the phase difference of a signal at the different sensors. Linear combination of the filtered signals can exploit the differences in sound level.

Similarly to the frequency response of a linear filter, one can define the direction-frequency response of spatio-temporal filter, i.e., beamformer, as the beamformer's response to a plane wave originating from a given direction. For simplicity, let us assume that signal source and microphones are located in a plane and that the phase of the signal at the first microphone is zero. If the signal $s(t) = e^{i\omega t}$ emanates from an angle θ to the microphone array and the time delay between the first and the m-th microphone is given as $\Delta_m(\theta)$, then the beamformer output is given as:

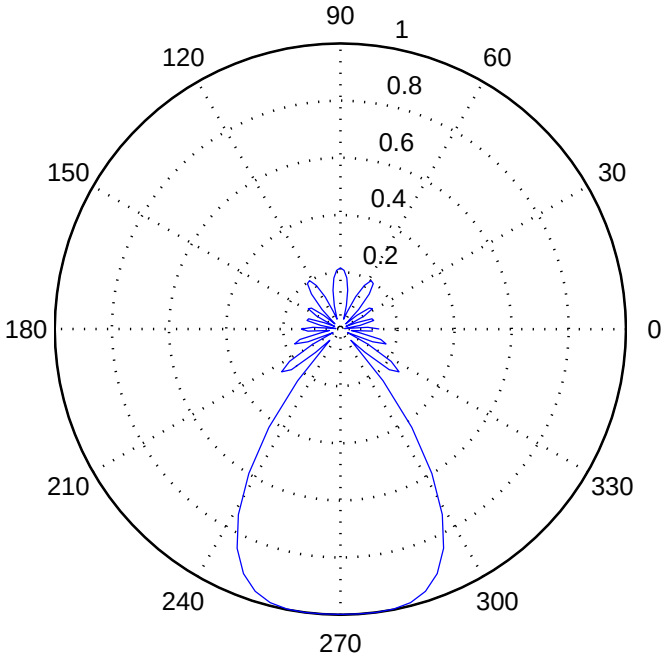


Figure 4.2: The beampattern $|r(\theta, \omega)|^2$ of a sum-and-delay beamformer.

$$b(t) = e^{i\omega t} \sum_{m=1}^{N} \sum_{k=0}^{L-1} w_{mk} e^{-i\omega(\Delta_m(\theta)+k)} =: e^{i\omega t} r(\theta, \omega).$$

The function $(\theta, \omega) \mapsto |r(\theta, \omega)|^2$ is called the beampattern. It is most easily interpreted if plotted for a fixed frequency ω_0. In Figure 4.2, the beampattern of a so-called *sum-and-delay beamformer* is given. The coefficients w_{mk} for this beamformer are chosen such that they correspond to the time shift $\Delta_m(\theta)$ for a given angle θ.

In order to benefit from beamforming when the steering direction is not known a priori it is necessary to find the source direction prior to beamforming. Several approaches to source localisation in the array processing model have been proposed. Most of them assume the following signal model:

$$r(t) = As(t) + n(t).$$

Here, the signals r, with $r(t) \in \mathbb{C}^N$, received by the microphone array are modelled as a linear combination $A \in \mathbb{C}^{N \times P}$ of the source signals s, with $s(t) \in \mathbb{C}^P$, and additive noise signals n, with $n(t) \in \mathbb{C}^N$. The matrix A is determined by array geometry and signal propagation. Each row of A is a vector describing how much the corresponding source signal is attenuated at each of the array elements. These vectors are called *steering vectors*. In this context, signals are usually modeled in the complex domain to account for receivers or signal models leading to complex valued signals.

Many source localisation strategies depend on estimations of the time-difference of arrival (TDOA) of the source signals at the array segments. In noiseless situations with only one

source signal, the time-difference of arrival can be estimated from cross correlations of the signals recorded at pairs of sensors. The so-called *generalised cross correlation* [KC76] extends this method by filtering the signals prior to correlation. If filtering is performed in such a way that those portions of the signals which have the highest signal-to-noise ratio are retained, TDOA estimation can be greatly improved in noisy situations. Correlation methods, however, are very difficult to apply in multi-source problems.

Subspace methods form a very popular family of approaches to the source localisation problem, especially in the multi-source case. They exploit the eigenstructure of the spatial covariance matrix[1] $R := E[r(t)r(t)^H]$ of the received signals. Assuming that the noise is uncorrelated to the signals, the covariance matrix R can be described by applying the signal model for r as

$$R = AR_sA^H + R_n$$

because of the linearity of the expectation operator. Here, $R_s = E[s(t)s(t)^H]$ and $R_n = E[n(t)n(t)^H]$ are the spatial covariance matrices of the signal and the noise, respectively. If R_s and A are of full rank, AR_sA^H has positive eigenvalues $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_P$. If we assume the noise to be spatially white, i.e., $R_n = \sigma^2 I$, where σ^2 is the noise variance, the eigenvalues of R are $\alpha_1 + \sigma^2, \ldots, \alpha_P + \sigma^2, \sigma^2, \ldots, \sigma^2$. Thus, the number of sources can be estimated from R by estimating σ^2. The vector space spanned by the eigenvectors belonging to the first P eigenvalues of R is called the *signal subspace*, its complement spanned by the remaining $N - P$ eigenvectors is called the *noise subspace*. Now, eigenvectors e from the noise subspace correspond to zero eigenvalues of AR_sA^H, i.e., $AR_sA^He = 0$. As a consequence, $A^He = 0$, i.e., steering vectors lie in the orthogonal complement of the noise subspace, hence in the signal subspace. This observation is the basis for subspace methods such as the *Multiple Signal Classification* algorithm (MUSIC) [Sch86]. Starting from a parametrisation of the steering vectors by the source bearings, a criterion measuring the length of the projection of steering vectors to the noise subspace allows to find source bearings by search-based or algebraic methods.

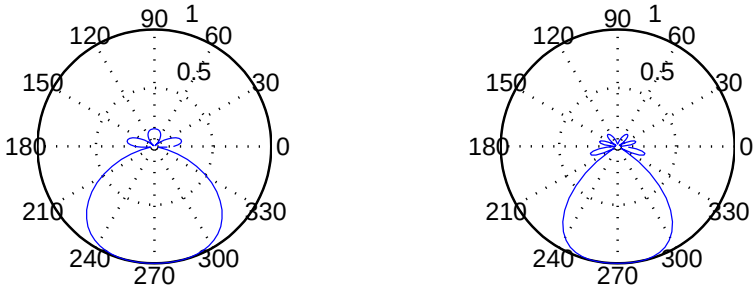
Another simple approach to source localisation is to steer a beamformer to various angles and compare the beamformer outputs. Together with a measure for the utility of such an output, optimisation over the space of admissible angles gives an algorithm for finding the direction of arrival. One example for such a measure of utility would be to steer the beamformer towards the direction with the loudest signal. This idea is one of the inspirations for the algorithm described in Section 4.3.

Beamforming comes with one main disadvantage in the case of complex audio scenes. This disadvantage is the dependence of the directivity pattern of a beamformer on the frequency of the source signal. In Figure 4.3, we give an example for the frequency dependence of the beampattern. One approach to overcome this problem is to use a hierarchical array architecture. This allows to use different sub-arrays for different frequency bands. This method, however, results in the need of a large number of microphones.

In addition to this problem, microphones have to be positioned quite carefully in order to apply beamforming methods. In Section 4.3 we therefore derive a method for source separation applicable to situations where few, more or less arbitrarily positioned, microphones are available.

[1] In this chapter, $E[x(t)]$ denotes the expected value of the random variable x. For vectors and matrices, ^{H} denotes the combination of complex conjugation and transposition.

a) $\omega_0 = 150Hz$ b) $\omega_0 = 250Hz$



c) $\omega_0 = 350Hz$ d) $\omega_0 = 450Hz$

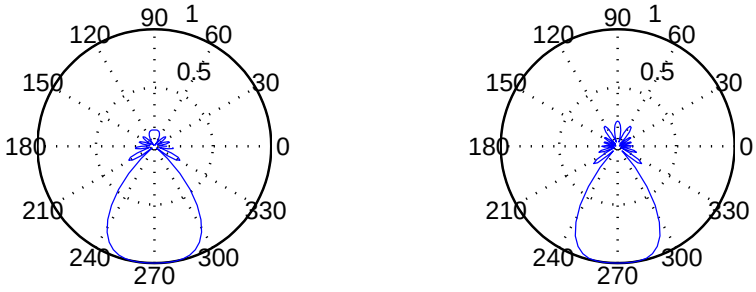


Figure 4.3: Frequency dependency of the beampattern of a sum-and-delay beamformer.

4.2 Independent Component Analysis

Another well-studied approach to the source separation problem is independent component analysis (ICA) [HKO01]. It starts with the observation that physically distinct audio sources should lead to statistically independent signals. Together with the assumption of linear mixture of sound sources it already yields algorithms for separation. Let us assume, we receive n signals which are linear mixtures of n sources:

$$m(t) = As(t).$$

Here, $m \in \ell^2(\mathbb{Z})^n$ are the received mixtures, $s \in \ell^2(\mathbb{Z})^n$ are the sampled source signals, and $A \in \mathbb{R}^{n\times n}$ is a time independent mixture matrix. If A would be known and of full rank, the source signals could be obtained from the mixtures by inverting A.

Understanding the values $s(t)$ as samples from a random vector s, we can model the mixture signals by a generative statistical model as the linear transform $m = As$ of the source random variables. The source variables s are assumed to be mutually statistically independent. Now, ICA algorithms try to find an approximation W of the inverse of A by maximising a measure of the statistical independence of the components of the unmixed sources u given by

$$u = Wm = WAs.$$

Statistical independence of the signals s is not affected by uniform scaling of each signal s_i by a factor α_i. Thus, absolute signal levels cannot be deduced by independent component

analysis. In order to remove the degree of freedom introduced by scaling, the search space for the unmixing matrix W can be restricted to matrices with determinant ± 1.

In [Com94] Comon examines the question of identifiability of the sources from the mixtures given the above model. It can be shown, that the sources are deducible from the mixtures if A is invertible and at most one of the source signals follows a Gaussian distribution. It is easy to see that two Gaussian signals mixed by an orthogonal mixture matrix follow a joint Gaussian distribution and thus lead to Gaussian mixtures which, in general, will always be independent.

In the model above, it has been assumed that the number of sources and mixtures are equal. If this is not the case, the problem becomes much more difficult. Also, the mixing matrix A is required to be of full rank. Otherwise, some of the sources are redundant and we are again facing the case where the number of sources is different from the number of mixtures.

Measuring statistical independence is a difficult task. This can be seen by considering the question which restrictions on moments or cumulants of random variables relate to statistical independence. The cross-cumulant $\operatorname{cum}(X_{i_1}, \ldots, X_{i_k})$ of a subset of n random variables is defined as the coefficient of the term $\omega_{i_1} \cdots \omega_{i_k}$ of the Taylor expansion of the characteristic function $\varphi(\omega) := E[\exp(i \sum \omega_j X_j)]$. One method for the preparation of statistical data is decorrelation up to order two. I.e., given a set of random variables, a linear transform is found that leads to random variables with vanishing covariance. Decorrelation up to order two is equivalent to finding statistically independent random variables in the Gaussian case. In general, however, statistical independence is equivalent to the vanishing of all cross-cumulants. Thus, no conditions on finite sets of joint moments or cross-cumulants can guarantee statistical independence. It is, however, possible to base approximate ICA algorithms on cumulant measures. Most prominently, the *Joint Approximate Diagonalization of Eigenmatrices (JADE)* algorithm [Car97] performs source separation by jointly diagonalising all fourth order cumulant tensors of the data. The JADE algorithm belongs to a class of ICA algorithms known as *algebraic algorithms*.

Other ICA algorithms combine approximations of statistical independence with local optimisation algorithms. As an example, we will consider the FastICA algorithm [HO97]. Here, independence is approximated by the negentropy J, which, for a random vector X is given as

$$J(X) = H(X_{\text{gauss}}) - H(X).$$

Negentropy is a normalised version of the differential entropy H, where normalisation is performed relative to X_{gauss}, which is a Gaussian random vector with the same covariance matrix as X.

Why negentropy is a good measure for statistical independence becomes clear from its connection to mutual information. The *mutual information* of random variables $X_1, \ldots, X_n$ is defined as

$$I(X_1, \ldots, X_n) := \sum_{i=1}^{n} H(X_i) - H(X).$$

Here, $X := (X_1, \ldots, X_n)$. An equivalent formulation of mutual information as the Kullback-Leibler divergence of the joint density of the X_i and the product of the marginal densities of the X_i shows that the mutual information of n random variables is zero iff the variables are statistically independent.

The relation of negentropy and mutual information is as follows:

$$I(X_1, \ldots, X_n) = J(X) - \sum_{i=1}^{n} J(X_i) + \frac{1}{2} \log \frac{\prod C_{ii}(X)}{\det C(X)}, \tag{4.1}$$

where $C(X)$ is the covariance matrix of X. Now, the third term in (4.1) is zero for uncorrelated X_i. Moreover, it can be shown that $J(AX) = J(X)$ for invertible linear transforms A [Com94]. Thus, for uncorrelated input signals m_i, an ICA algorithm can be formulated as the minimisation of $I(Wm_1, \ldots, Wm_n)$, which is equivalent to the maximisation of the functions $W \mapsto J(Wm_i)$.

In [Hyv98], Hyvärinen gives approximations of negentropy by non-polynomial functions. For a nonlinear function G, negentropy can be approximated as

$$J(x) \approx c(E[G(x)] - E[G(\nu)])^2 =: \tilde{J}(x)$$

for a constant c and a zero mean, one-dimensional Gaussian distribution ν with unit variance. In practice, functions like $G(x) = \frac{1}{a} \log \cosh ay$ have proven useful.

Now, if w_i is the i-th row of the unmixing matrix W, based on the gradient of the approximation $\tilde{J}(w_i s_i)$, a fixed point formulation of the optimisation problem can be found which can be solved by an approximative Newton method. It does however require renormalisation of the vector w after each iteration in order to guarantee that the resulting matrix W has determinant ± 1.

Classical optimisation algorithms are not very well adapted to the ICA task. This stems from the fact that optimisation is performed on matrix spaces which are not Euclidean. This is the reason why renormalistion has to be performed after each optimisation step in the FastICA algorithm. It is much more natural to respect the geometric nature of the parameter space. Therefore, much research is focused on *geometric algorithms* taking into account the structure of the underlying matrix spaces [Plu05].

Applying ICA to real-world audio data is a very difficult problem [HKO01]. The main reason for this is that the basic ICA model does not take into account the complex processes leading from sound generation at the sound sources to the mixtures recorded by microphones. These processes comprise effects like echoes, variable number of sources, variable mixing conditions, etc., as described in Chapter 1. In this setting, it can already be considered a success, if the utterances of a small number of speakers recorded by the same number of microphones can be separated. This can be achieved by ICA methods, for example, when special attention is given to the estimation of time delays [LZOS98]. In the next section, we develop an algorithm for source separation which gives up the strict requirement to exactly recover source signals in order to achieve the extraction of simpler components in difficult mixing conditions.

4.3 Spectral Flatness Components

In very complex audio scenes, it is usually not feasible to reconstruct the single source signals. In this section, we therefore develop an algorithm that extracts combinations of the input signals that constitute less complex components of an audio scene. These components will be simpler to analyse than the complex mixture.

The central step in devising our source separation algorithm is the choice of a measure describing the complexity of an audio scene. Given such a measure, it is possible to evaluate

it for several combinations of input sounds and choose the combination that gives the lowest complexity score.

The measure we use in our approach is the spectral flatness measure. It measures how much the energy at a given time is spread in the spectrum, giving a high value when the energy is equally distributed and a low value when the energy is concentrated in a small number of narrow frequency bands. The spectral flatness measure is computed from the spectrum as the geometric mean of the Fourier coefficients divided by the arithmetic mean. If $S(\omega, t)$ is the windowed power spectrum of a signal s, its *spectral flateness measure* is given by

$$\mathrm{S}FM[S](t) = \frac{(\prod_{\omega=0}^{\Omega-1} S(\omega,t))^{\frac{1}{\Omega}}}{\frac{1}{\Omega}\sum_{\omega=0}^{\Omega-1} S(\omega,t)}.$$

Some care has to be taken in applying this definition. Firstly, if $S(\omega, t) = 0$ for one value of ω, $SFM[s](t)$ is zero, too. Therefore, we replace each value $S(\omega, t)$ by $S(\omega, t) + \epsilon$ for some small value $\epsilon > 0$. Secondly, direct computation of the numerator is numerically unstable. This problem is overcome by regarding that $(\prod_{\omega=0}^{\Omega-1} S(\omega,t))^{\frac{1}{\Omega}} = \prod_{\omega=0}^{\Omega-1}(S(\omega,t)^{\frac{1}{\Omega}})$.

The spectral flatness measure is also known as *Wiener entropy* and has already been used for measuring the similarity of bird songs [TNH$^+$00, DMF$^+$05].

Given a sequence of signals $f := (f_1, \ldots, f_n)$ from a microphone array we assume that a lower complexity source can be derived by choosing a linear combination of the signals. In order to apply the spectral flatness measure, we are interested in the windowed power spectrum U_f of such a linear combination:

$$U_f(\omega, t; a_1, \ldots, a_n) := |\sum_{i=1}^{n} a_i \hat{f}_i(\omega, t)|^2.$$

In the same way in which scaling does not affect statistical independence, it also does not change the complexity measure of a component U_f. Hence, we assume $\|(a_1, \ldots, a_n)\|_2 = 1$.

The mixing coefficients of the source signals should be estimated from segments of constant mixing conditions. We assume that mixing conditions are locally constant and form a windowed spectral flatness measure by convolving short signal windows with a Hann window. If $h(n) := \frac{1}{2}(1 - \cos(\frac{2\pi(n-W)}{2W}))$ denotes a discrete Hann window of length $2W + 1$ centered at 0, the measure of complexity Φ for the mixture U_f is given by:

$$\Phi[U_f(\cdot,\cdot; a_1, \ldots, a_n)](x) := \sum_{t=-W}^{W} h(t)\mathrm{SFM}[U_f(\cdot,\cdot; a_1, \ldots, a_n)](x + t).$$

Figure 4.4 gives an example of the spectral flatness measure of an audio signal and the windowed measure. In the following we will first give an overview of how to extract lower complexity components based on this complexity measure. Afterwards, we will give a more detailed description of the algorithm, followed by experimental results for artificial as well as natural audio scenes.

Starting from p initial hypotheses $A \in \mathbb{R}^{p\times n}$ for the unmixing coefficients at the first analysis window of the signal, starting with time position τ_0, we use an optimisation algorithm to find local minimisers $(a_{k,1}, \ldots, a_{k,n})$ of $\Phi[U_f(\cdot,\cdot; a_1, \ldots, a_n)](\tau_0)$. Because of our assumption that $\|(a_1, \ldots, a_n)\|_2 = 1$, we represent each hypothesis k — given by row k of A — by polar coordinates $H_{k,1}, \ldots, H_{k,n-1}$. Now, we examine the audio scene at T equally spaced time steps and estimate p hypotheses for each of them by applying a local optimisation algorithm

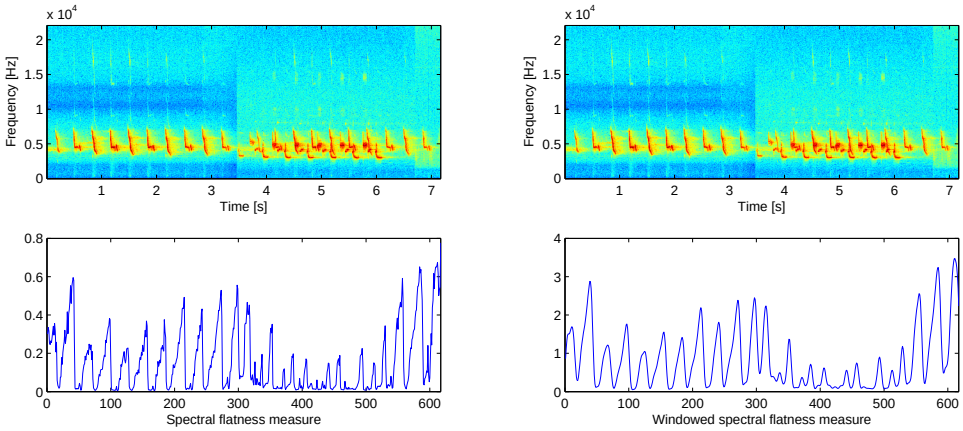


Figure 4.4: Left: The spectral flatness measure of a mixture of bird songs. Right: Windowed version of the measure which is used as an objective function for source separation.

starting from the hypotheses of the previous time step. In this way, we enable our algorithm to start from good initial hypotheses whenever the unmixing parameters vary slowly. Therefore, our algorithm performs source tracking where applicable. Finally, we find hypotheses $H \in \mathbb{R}^{p \times n-1}$ for each time step. These can be combined in a matrix $\tilde{H} \in \mathbb{R}^{p(n-1) \times T}$. In order to extract meaningful components, we apply a dynamic programming algorithm to the matrix $\tilde{H}$ which extracts up to p components by selecting one hypothesis for each time step. In each component, the algorithm minimises the ℓ_2-distance between mixture hypotheses of neighbouring time steps. Moreover, we ensure, that no single mixing hypothesis is used in more than one extracted component. Mixing coefficients for time positions between two time steps are found by linear interpolation.

The Algorithm

In order to complete the algorithm sketched above, three parts have to be detailed. First, we need a sensible initialisation of the hypotheses for the mixing coefficients of the first time step regarded by the algorithm. Second, we use gradient-based optimisation to find good mixing coefficients and therefore need to compute the partial derivatives of the complexity measure w.r.t. the mixing coefficients. Finally, we need to choose components from the hypotheses generated by the optimisation procedure.

Initial Hypotheses

We start by generating h initial hypotheses for the mixing parameters. Each hypothesis, given by polar coordinates $H_{k,1}, \ldots, H_{k,n-1}$, describes a point on the sphere S^{n-1}. Therefore, a good covering of the parameter space is obtained by choosing h points on the sphere which are distributed as regularly as possible. Finding a good measure for the regularity of the distribution is far from trivial. A number of measures have been proposed emphasising different aspects of the point distribution (see, e.g., [CSB87]). A very simple approach is to find approximate solutions to Thomson's problem [Tho04]. This problem asks for the distribution of a fixed number h of electrons on a sphere, when the electrons repel each other

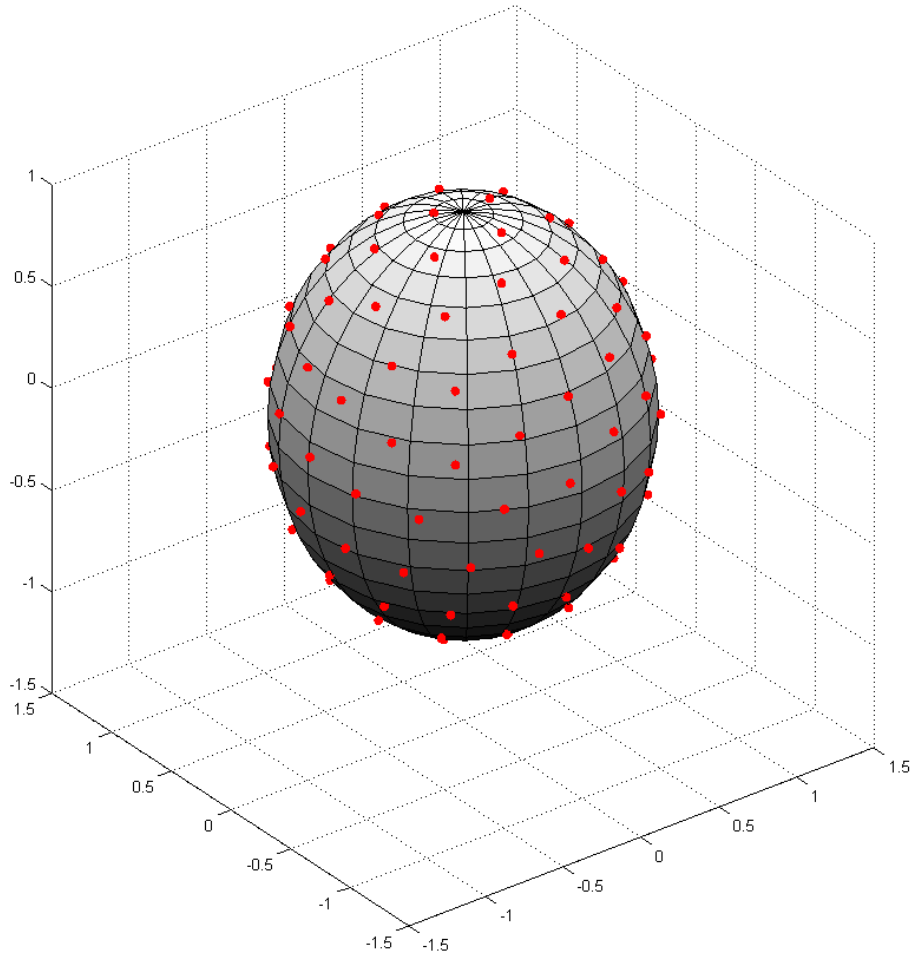


Figure 4.5: An approximate solution to Thomson's problem resulting in a "regular" distribution of 117 points on a sphere.

following Coulomb's law. Two equal electric charges repel each other with a force inversely proportional to the square of their distance.

An approximate solution (see Figure 4.5 for an example) is found by iteratively moving each point according to the sum of the forces exerted by the other points. This simple approach gives a good distribution of the points for the initial hypotheses used in our algorithm after a small number of iterations. For the numbers of hypotheses used in our algorithm, the computational cost for this initialisation process is negligible.

Optimisation

In order to use the complexity measure Φ defined above in an optimisation algorithm, we need to compute the partial derivatives with respect to the mixture weights. The mixture weights $(a_1, \ldots, a_n)$ are given as functions of their polar coordinates $(\varphi_1, \ldots, \varphi_{n-1})$ as follows:

$$a_s = \begin{cases} \cos(\varphi_s) \prod_{i=1}^{s-1} \sin(\varphi_i) & s < n \\ \prod_{i=1}^{n-1} \sin(\varphi_i) & s = n \end{cases}.$$

In the following, we will need the derivatives of the weights with respect to the polar coordinates. For a_n they are given as

$$\frac{\partial a_n}{\partial \varphi_j} = \cos(\varphi_j) \prod_{i \neq j} \sin(\varphi_i).$$

For a_s, $s < n$, the derivatives are

$$\frac{\partial a_s}{\partial \varphi_j} = \begin{cases} \cos(\varphi_s)\cos(\varphi_j)\prod_{i=1, i\neq j}^{s-1} \sin(\varphi_i) & j < s \\ -\prod_{i=1}^{s-1} \sin(\varphi_i) & j = s \\ 0 & j > s \end{cases}.$$

In order to simplify the notation, in the following, we will abbreviate

$$(a_1(\varphi_1, \ldots, \varphi_{n-1}), \ldots, a_n(\varphi_1, \ldots, \varphi_{n-1})) =: a(\varphi).$$

Now, we can proceed to calculating the partial derivatives of $\Phi[U_f(\cdot, \cdot; a(\varphi))]$ with respect to the polar coordinates $\varphi := (\varphi_1, \ldots, \varphi_{n-1})$:

$$\frac{\partial \Phi}{\partial \varphi_j}(x; \varphi) = \sum_{t=-W}^{W} h(t) \frac{\partial}{\partial \varphi_j} SFM[U_f(\cdot, \cdot; a(\varphi))](x + t).$$

Thus, in order to compute the partial derivatives of Φ, we need to know the partial derivatives of SFM. For this purpose, we abbreviate numerator and denominator of SFM by $p(x; \varphi) := \prod_{\omega=0}^{\Omega-1} (U_f(\omega, x; a(\varphi)))^{\frac{1}{\Omega}}$ and $q(x; \varphi) := \frac{1}{\Omega} \sum_{\omega=0}^{\Omega-1} U_f(\omega, x; a(\varphi))$, respectively. Now, the derivative is

$$\frac{\partial SFM[U_f(\cdot, \cdot; a(\varphi))]}{\partial \varphi_j}(x; \varphi) = \frac{\frac{\partial p}{\partial \varphi_j}(x; \varphi) q(x; \varphi) - p(x; \varphi) \frac{\partial q}{\partial \varphi_j}(x; \varphi)}{q^2(x; \varphi)}.$$

Here, the corresponding derivatives of p and q are given by

$$\frac{\partial q}{\partial \varphi_j}(x; \varphi) = \frac{1}{\Omega} \sum_{\omega=1}^{\Omega} \frac{\partial}{\partial \varphi_j} U_f(\omega, x; a(\varphi))$$

and

$$\frac{\partial p}{\partial \varphi_j}(x; \varphi) = \frac{1}{\Omega} p(t) \sum_{\omega=1}^{\Omega} \frac{\frac{\partial}{\partial \varphi_j} U_f(\omega, x; a(\varphi))}{U_f(\omega, x; a(\varphi))}.$$

Finally, the derivative of U_f can be computed from the data as

$$\frac{\partial U_f(\omega, x; a(\varphi))}{\partial \varphi_j} = \sum_{i=1}^{n} \hat{f}_i(\omega, x) \frac{\partial}{\partial \varphi_j} a_i(\varphi_1, \ldots, \varphi_{n-1}).$$

Using this derivative, locally optimal unmixing parameters can be found by a variant of a line-search algorithm. We found the simple Algorithm 1, adjusting the step size by a factor of two according to whether or not the new step size gives a better increase in the objective function than the last one, to be sufficient for our purposes.

Algorithm 1 A simple line-search step for the optimisation of U_f.

Require: step size α from previous optimisation step, objective function U_f, position x and derivative dx.

```
state := 0
done := false
while not done do
  v := U_f(x + α · dx)
  if v > U_f(x) then
    if state == 2 then
      done := true
    else
      state := 1
    end if
    α := 2α
  else
    if state == 1 then
      done := true
    else
      state := 2
    end if
    α := α/2
  end if
end while
return x + α · dx
```

Component Extraction

Finally, after transforming back angular representations into unmixing parameters, we have computed a matrix $H \in \mathbb{R}^{hn \times T}$ giving, at each time step, h hypotheses for the mixing parameters. A component is described by a sequence $((h_1, 1), (h_2, 2), \ldots, (h_T, T))$. To each element (i, j) of such a sequence, we associate the corresponding vector $v(i, j) := (H(i, j), \ldots, H(i + n - 1, j))$ of unmixing parameters. In order to extract components without abrupt changes in the unmixing parameters, where possible, we want to minimise the sum S of differences between the unmixing parameters for each component:

$$S((h_1, 1), \ldots, (h_T, T)) := \sum_{i=1}^{T-1} \|v(h_{i+1}, i + 1) - v(h_i, i)\|$$

Moreover, we want to extract disjoint components, i.e., each parameter vector $v(i, j)$ should appear in at most one component.

The first component can be extracted by computing the values of matrices D and P defined as follows. Entry D_{ij} gives the minimal costs for a component of length j, ending with the parameter vector $v(i, j)$. Entry P_{ij} gives the predecessor of $v(i, j)$ on a minimal cost component. D can be computed by standard dynamic programming, observing that $D(i, 1) = 0$ for all i and

$$D(i, j) = \min_k D(k, j - 1) \text{ for } j > 1. \tag{4.2}$$

The entries of P are easily found by storing the minimisers in (4.2).

In order to guarantee disjoint components, this process has to be slightly modified after the extraction of the first component by setting $D_{ij} = \infty$ whenever $v(i, j)$ has already been used in a component.

For the analysis of audio scenes, it is usually sufficient to reconstruct the spectrum of a component by applying the selected unmixing parameters. It is, however, sometimes desirable to reconstruct audio signals from the components. In this case, it is possible to use the estimated power spectrum as a mask for weighting the complex spectrum of the mixed signals. Then, a signal can be reconstructed from this weighted spectrum which includes phase information in addition to the magnitude information contained in the power spectrum.

Results

The algorithm described above has been tested on a variety of datasets. First, in order to test basic functionality, artificial mixtures of natural sound sources have been created. Then, in order to get closer to real-world applications, we present an example of two speakers recorded by two microphones in an office room. Finally, we will show results on real-world monitoring recordings of animals recorded by four microphones.

Artificial mixtures of sound sources were created using recordings from three species of birds, the chiff-chaff, the great tit and the chaffinch as depicted in Figure 4.6.

In the first test, the songs of chiff-chaff and great tit were mixed with constant mixing parameters to obtain the signals shown in the top of Figure 4.7. The first mixture is obtained by weighting each signal by a factor of $\sqrt{2}$, the second signal is obtained by weighting the great tit song by a factor of $\frac{\sqrt{3}}{2}$ and the chiff-chaff song with a factor of $\frac{1}{2}$. As can be seen in the figure, the chiff-chaff song can be recovered almost perfectly from these mixtures.

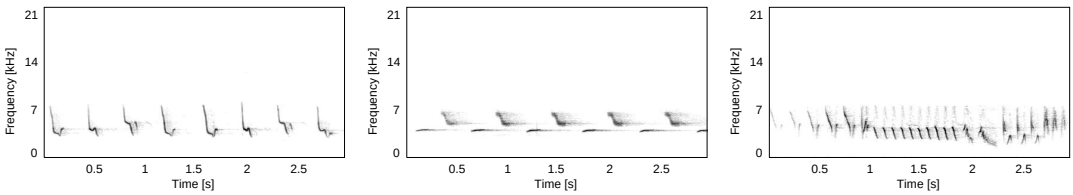


Figure 4.6: Three signals used for artificial mixing of test signals: Chiff-chaff (left), great tit (middle) and chaffinch (right).

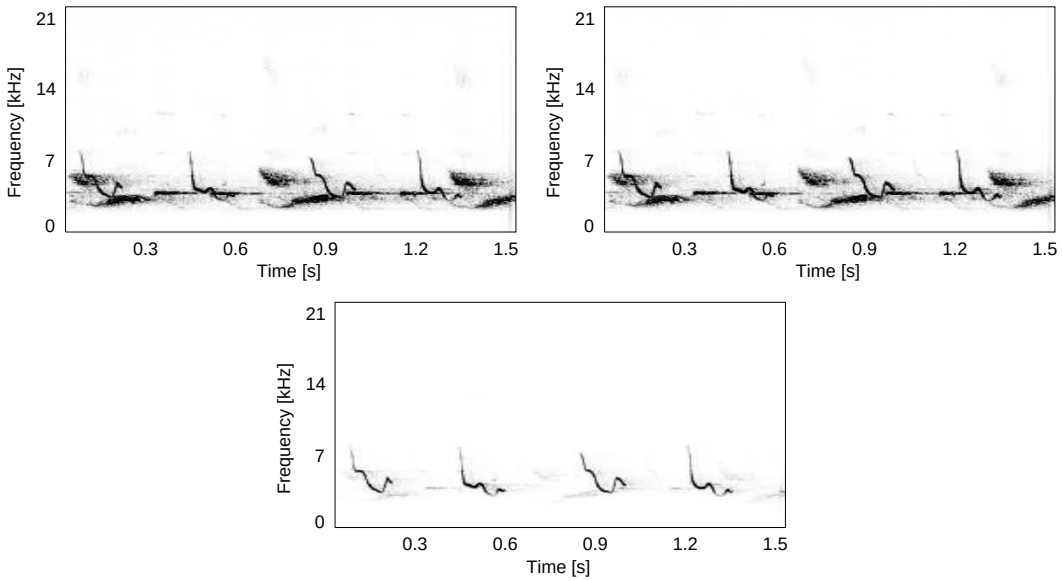


Figure 4.7: Source separation for two artificially mixed sources. The upper part shows the spectrograms of two different mixed signals of chiff-chaff and great tit song. The lower part shows the spectrogram of the simplest component extacted from these mixtures.

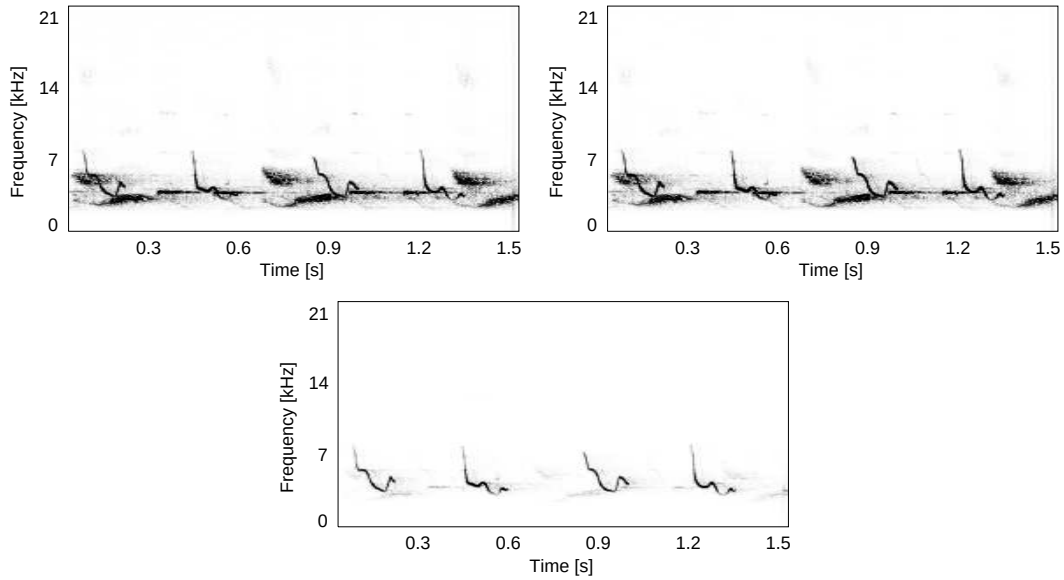


Figure 4.8: Source separation for two artificially mixed sources with varying mixing parameters.

One of the main features distinguishing our method of source separation from previous solutions is that it is able to track varying mixing conditions. Figure 4.8 shows an example of the same two signals as in the previous example, this time mixed by varying mixing parameters. For a second mixture, the two source signals were mixed by weighting the first one by the factor $\sin\alpha$ and the second one by the factor $\cos\alpha$ for an angle α which changes gradually over the playback time of the signals. It starts with $\alpha = 0$ in the first fourth of the signal, then changes to $\frac{\pi}{8}$ in the second fourth, to $\frac{\pi}{4}$ in the third, and to $\frac{3\pi}{8}$ in the final fourth. A second mixture signal was obtained by using the weight $\cos\alpha$ on the first source signal and the weight $\sin\alpha$ on the second. Mixing parameters are interpolated linearly between the different parts. Again, the song of the chiff-chaff can be extracted from the mixtures with the same quality as in the case of constant mixture parameters.

As a final test with artificial mixtures, we present a result for a mixture of three sources with varying mixing parameters. The first three spectrograms in Figure 4.9 show three mixtures of the three signals shown in Figure 4.6. The mixing parameters have been created similar to those in the previous example. Using the same angle α as a parameter, the signals have been mixed according to Table 4.1. Here, the song of the chaffinch is extracted almost perfectly, except for some remnants of the song of the great tit to be found at the beginning and the end of the extracted component.

A result which is closer to real world applications is the separation of a mixture of two speakers talking simultaneously in a room **recorded by two microphones**. The data used for this test is an example from an ICA-based source separation method which incorporated estimating time delay between the two microphones [LZOS98]. Note, that no estimation of time delay is necessary for our method. Figure 4.10 shows spectrograms of the signals recorded by the microphones. Each of the two speaker is counting from one to ten, one speaker in English, one in Spanish. Their utterings strongly overlap in time and frequency. Looking at the spectrograms shows that the extracted component is indeed much simpler

Mixture	Weight for signal 1	Weight for signal 2	Weight for signal 3
Mixture 1	$\cos(\alpha)$	$\sin(\alpha)$	$\cos(\alpha + \frac{\pi}{8})$
Mixture 2	$\sin(\alpha + \frac{\pi}{8})$	$\cos(\alpha)$	$\cos(\alpha)$
Mixture 3	$\cos(\alpha + \frac{\pi}{8})$	$\cos(\alpha)$	$\sin(\alpha)$

Table 4.1: Mixture parameters for the three signals shown in Figure 4.9. The mixtures were created using an angle α varying from 0 over $\frac{\pi}{8}$ and $\frac{\pi}{4}$ to $\frac{3\pi}{8}$.

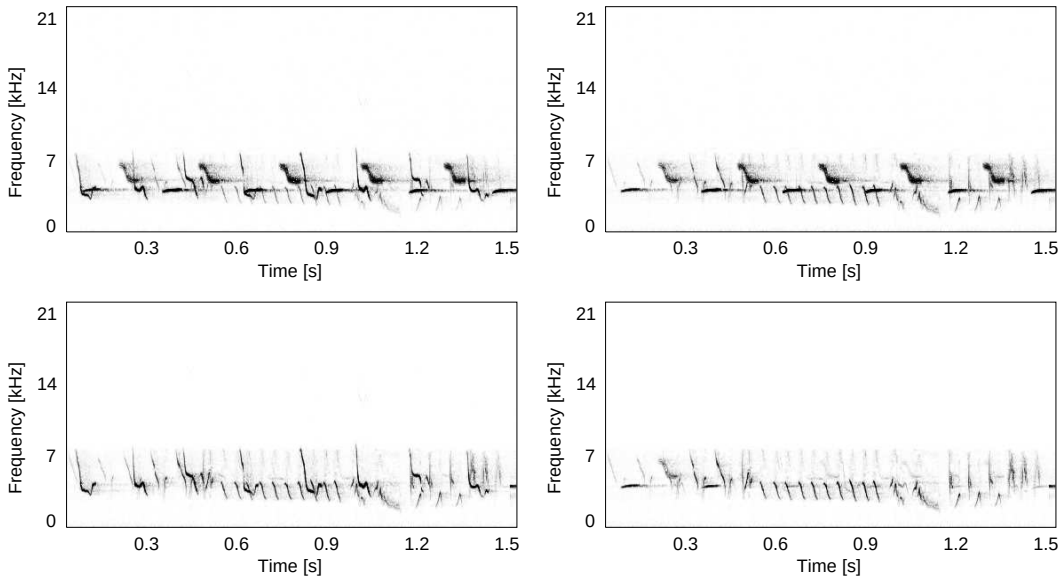


Figure 4.9: Source separation for three artificially mixed sources with varying mixing parameters.

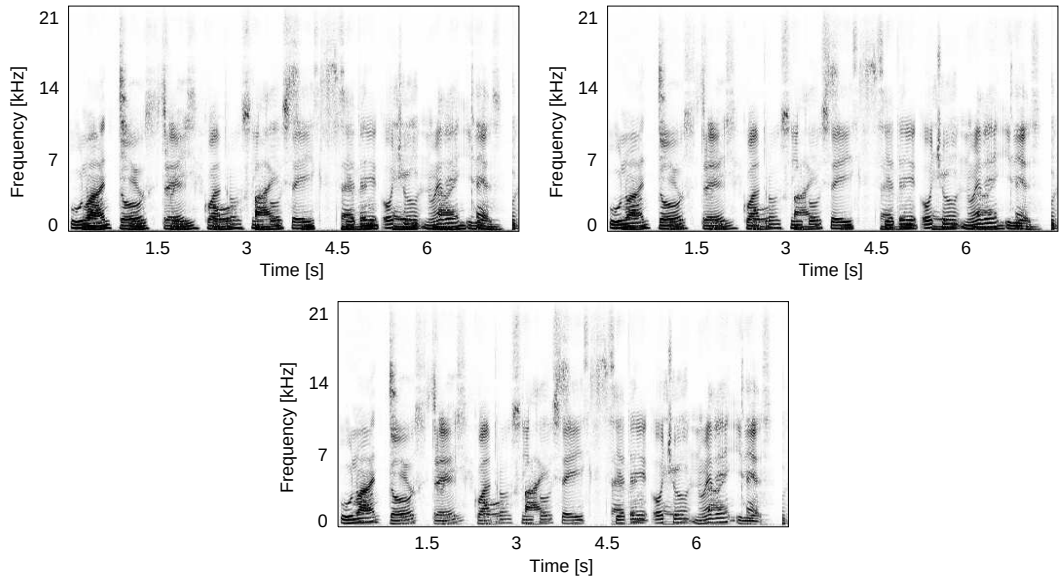


Figure 4.10: Source separation for two speakers in a room recorded by two microphones. The upper spectrograms show spectrograms of the two signals picked up by the microphones. The lower spectrogram shows the best component reconstructed by the algorithm.

than the mixtures. Listening to a reconstructed signal reveals that the utterings of one of the speaker are strongly attenuated.

Finally, we used real-word **monitoring recordings** in order to find out whether simplifications of such recordings can be obtained by our method. Figure 4.11 shows an example of an audio scene recorded with three microphones. It contains the vocalisations of a number of animals, for example the song of Savi's warbler (see Chapter 5). For such complex audio scenes, our method is not yet very successful. Still, a simplification of the scene can be observed. For example, the bird vocalisation at about 4kHz from 0.2s to 0.4s is clearer discernable in the extracted component than in any of the mixture signals.

Unfortunately, usually only the best component extracted from the mixtures gives sensible results. The other components extracted are very similar to the best component most of the time. Thus, in order to extract more than one component with our methods, a different way of extracting these additional components from the optimisation hypotheses has to be found. A simple method to do so would be to subtract the best extracted component from the mixtures with a suitable weight and then repeat the optimisation process with these simplified mixtures. Another idea would be to include a measure of similarity to already extracted components into the component extraction process.

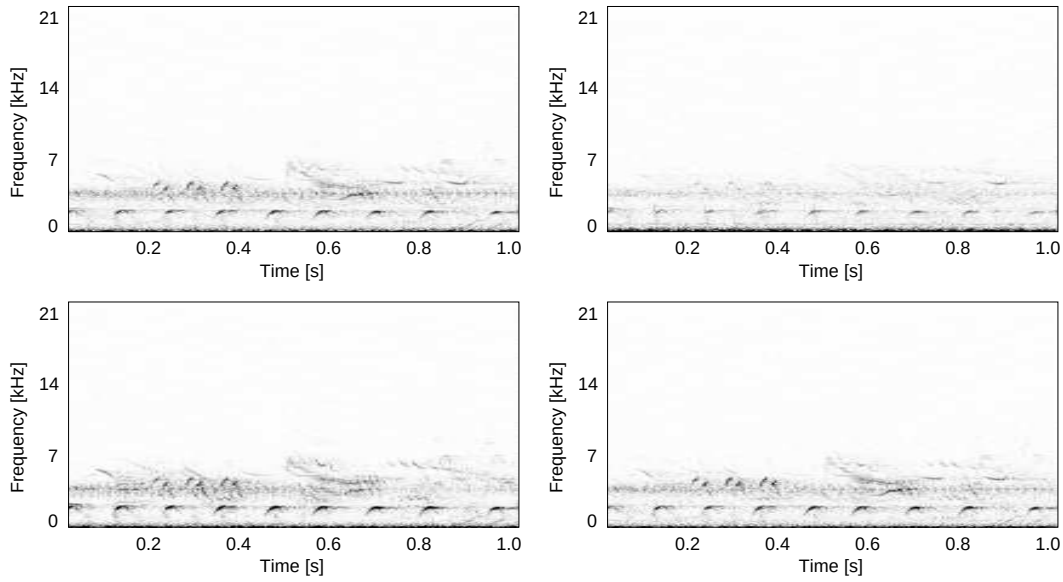


Figure 4.11: Source separation for a real-world monitoring recording. Three signals recorded by microphones are depicted in the upper and the lower left spectrograms. The lower right spectrogram shows a component extracted with our algorithm.

Chapter 5

Algorithms for Animal Species Recognition

In the previous chapter, we have dealt in depth with the problem of breaking down an audio scene into simpler parts. These will now be subject to closer analysis. In this chapter we will turn our attention towards the problem of finding time segments in audio recordings where a given species of animal is audible. Although less work has been done concerning this task, as compared to other tasks in pattern recognition, such as speech recognition, a variety of methods and animal species have been examined, and, in Section 5.1, we will first give an overview of such related work.

The task of detecting and recognising vocalisations of animal species in continuous audio recordings is of great interest in supporting monitoring programmes for nature conservation. Here, the efficacy of conservation action is measured by observing the development of population sizes of certain target species. They are an indicator for the health of habitats. Assessing these numbers is currently performed by human listeners and is highly laborious. In Section 5.2, we give an overview of a project examining how pattern recognition algorithms can support, alleviate and objectify this task. The algorithms described in this chapter have been developed in the context of this project and were evaluated using recordings from the project.

In this context, it is often desirable to monitor for the presence of a small number of especially interesting target species. We will describe algorithms for three such target species in Section 5.3. Two of these species are interesting for conservational reasons, whereas the third species is prototypical for song birds with strongly structured songs. In designing special purpose algorithms for the recognition of single species, one hopes to identify principles that are applicable in the design of detectors for other species. We will therefore discuss the applicability of the methods developed in this chapter to the recognition of other animal species.

5.1 Related Work

In comparison to other fields in pattern recognition, little work has been carried out regarding animal sound recognition. Nevertheless, a wide variety of methods and animal species have been examined. Already in early studies, bird song recognition with hidden Markov models has been proved to be a useful tool in the recognition of bird song elements [KM98]. In this case, recordings were made under laboratory conditions with captive birds and microphones

close to the cages.

Almost all vocalising animal species conceivable have been studied. The most obvious candidates for species recognition by sound are birds. But a lot of other species have also been subject to efforts in automated recognition, for example, crickets and grasshoppers, marine mammals like whales and dolphins, frogs and bats. Such methods are of interest in applications such as monitoring for the presence of certain species in an area, behavioural studies, assessing the impact of anthropogenic noise on animal vocalisations and many more.

Currently, there is no recipe for the direct application of standard methods from machine learning to pattern recognition problems in time dependent data. The main problem here is how to decide which parts of a signal are to be used as input for such methods. Often, this problem is dealt with by applying segmentation algorithms. Unfortunately, this is equivalent to finding the starting and ending positions of animal vocalisations or their segments which is extremely difficult. This is particularly true for natural audio scenes where current solutions tend to be unreliable because of low signal-to-noise ratios. If, however, this gap is bridged in some way or the other, numerous classification algorithms are available for application. The classical dynamic time warping algorithm [DPH93] has found application in cases where animal vocalisations are not too variable and can thus be recognised by template matching [BM07]. Neural networks present an obvious candidate for pattern classification. In addition to bird calls [Mil00] and bird song elements [NBDS06], they have been applied to the recognition of animal species like marine mammals [DFS99] and crickets [SDK+03]. Self-organising maps have also been used for various classification tasks concerning animal sounds [SH03, MFW96, PSB+06]. Also, decision tree classifiers like C4.5 [Qui93] have been used for bird song recognition [Tay95].

When regarding spectrograms of animal voices, the idea of applying image analysis methods springs to the mind. In [OBF04], a synergetic pattern recognition algorithm [HTR98] is used to classify images of bat spectrograms. In [BNF06], more classical image processing tools, such as image filtering and thresholding, are used in order to extract features from the spectrogram.

In Chapter 3, we have already encountered animal vocalisations that are not very well represented using the Fourier transform. These vocalisations are noise-like in the sense that their energy spreads over a broad frequency range. Wavelets have been proposed as an alternative for analysing such sounds. They have been shown to concentrate energy in comparatively few wavelet coefficients for sounds which otherwise need many Fourier coefficients for their representation [STT07]. In [Pos02] and [SP03], wavelets have been applied as feature extractors for the classification of bird and whale songs. In the latter case improvements over spectrogram matching techniques have been achieved. Here, wavelets were also introduced as a means to tackle the segmentation problem described above.

One of the few projects directly dedicated to the recognition of bird songs is the Avesound project. It is a cooperative project of the universities of Tampere and Helsinki. It has so far led to two master theses [Fag04, Sel05]. We will now discuss some of the work conducted in this project. One of the main tools which has been used is self organising maps. Different methods of constructing such maps from bird songs have been examined [SH03] and a comparison of the maps for different bird species showed that the differences in the variability of the songs from different species were well reflected in the maps. Because of the tonal quality of many bird songs, sinusoidal modelling is a promising feature extraction step for bird song recognition and has been studied extensively in this context [Här03]. Somervuo et al. [SH04] deal with birds songs composed of a fixed set of syllables in variable composition. In order to create

fixed size input vectors for nearest neighbour classifiers, histograms of syllable pairs were computed. Starting with an isolated bird song, a histogram $h_{i,j}$ is formed by defining $h_{i,j}$ as the sum of all probabilities that syllable i is directly followed by syllable j in the song. The quality of such histograms obviously depends strongly on the quality of the syllable models giving the probability of the presence of a syllable at a given time. We have already commented on the problem of finding good features for transient sounds. In [FH05], other representations than wavelets have been proposed. For example, MFCCs have been found to give good representations in such cases. Moreover, some bird songs are rich in harmonic structure, a fact that can be used for their recognition [HS04]. More generally, several sets of features for the representation of bird songs have been compared within the project [SHF06]. Again, MFCCs have resulted in good fixed dimensional features and were also a good basis for the extraction of features based on frequency trajectories. Still, the recognition of bird calls in natural environments remains a great challenge [TTSO06]. Recently, recognition results could be improved by using support vector machines arranged in a decision tree [Fag07].

One area with comparatively high focus on automated recognition of acoustic patterns is underwater bioacoustics. Although many of the problems are the same as in conventional bioacoustics, underwater acoustics has some properties making it somewhat special. For example, high frequency sound does not carry very far in water. Thus, the environment behaves like a strong low-pass filter. On the other hand, low frequency sounds carry vast distances. In most cases, anthropogenic noise does not play as crucial a role in underwater bioacoustics as it does on land. Moreover, underwater audio scenes tend to be less complex because there are less species who are vocalising. Apart from the acoustic situation, the methods used for recognising species underwater do not vary from those used on land. For example, dynamic time warping [BM07], wavelets, self-organising maps [DFS99] and neural networks are applied. A recent overview of methods applied to the passive acoustic observation of cetaceans is given by Mellinger et al. [MSM$^+$07].

Another area with a special acoustic situation is the detection and recognition of bat vocalisations [OBF04]. The main difference to other situations is that special technical means have to be applied in order to capture the ultra-sound signals emitted by bats. Here, either frequency division detectors or recording devices with very high sampling frequencies are applied. Similar as for underwater audio scenes, noise and concurrent audio sources do not pose problems in the ultra-sound range as much as they do in the audible frequency range. The most challenging problem in the bat scenario is that often mainly echo-location calls can be recorded and these are quite similar between different species.

The calls of insects like crickets and grasshoppers have special structure making them comparatively easy to recognise. Often, they are characterised by highly regular repetition of simple click sounds. Schwenker et al. showed that sets of simple features like repetition frequency allow high species recognition rates on pre-segmented recordings of cricket songs using neural networks [SDK$^+$03]. A very interesting application of insect species recognition by sound is in the detection of pests in imported goods [FC07].

5.2 Bioacoustic Pattern Recognition

For the evaluation of our pattern recognition algorithms, we rely on recordings made at Lake Parstein in North Eastern Brandenburg in 2006 and 2007. These were conducted in a joint project with the Animal Sound Archive Berlin on *Bioacoustic Pattern Recognition*. The

recordings were made by the research group of the Animal Sound Archive. The project was funded by the German Federal Agency for Nature Conservation (BfN). Its aim was to examine the applicability of pattern recognition methods as a tool for monitoring bird vocalisations. The study area was chosen because it is home for threatened bird species and is difficult to access for humans. Therefore, automated monitoring methods can be extremely helpful in such regions.

A four-channel stationary microphone array of cardioic microphones was used on several positions at the lake shore, from a lookout, and from a boat. An autonomous recording station powered by solar panels was dispatched on a boat, see Figure 5.1. The four channel recordings were used in order to perform experiments with source separation and localisation.

Recordings were performed by night in order to cover audio scenes with a complexity level somewhere between the simple situation of a laboratory recording and the extremely complex situation of bird choirs at daytime. At the study site, two species of bittern and Savi's warbler were chosen as targets for monitoring. The noise level by night was unexpectedly high, especially due to the calls of amphibians such as tree frogs. Calls of the bittern could be recorded over distances of about 1 km. The best recordings of vocalisations from the reed zone were achieved when the microphones were placed on a boat on the lake.

One of the standard procedures for assessing the number of individuals of given species in an area is line mapping. Here, an observer moves along a predefined route — the line — with constant speed, making note of all bird species he notices. In order to assess the applicability of pattern recognition tools in line mapping, the lake shore was traversed by boat, recording the geographical coordinates with a GPS device. These recordings were evaluated in three ways, see Figure 5.2. First, an observer on the boat produced a map of the places where Savi's warbler was heard during the tour without technical means. Second, an experienced listener produced a similar map by listening to the recordings and using the GPS data. Listening to stereo recordings was used to help finding the direction from which the calls were coming. Finally, another map was produced using the GPS data and the pattern recognition algorithm described in Section 5.3.

More details on data acquisition and the less algorithmic parts of the project are found in a report of the group from the Animal Sound Archive [FTK08].

5.3 Special Purpose Algorithms

In this section, we describe algorithms designed for the purpose of detecting the presence of vocalisations of a small number of target species chosen for their value for nature conservation. In addition to these species, the recognition of chaffinch songs is investigated as an example for a strongly structured song with relatively high variability. The algorithms sketched in this section are joint work with Daniel Wolff and more details can be found in his thesis [Wol08]. We will first describe three different detectors and then deal with the applicability to other species.

The three types of special purpose algorithms described in this section are tailored for different types of signals. The first algorithm to be described will be useful for detecting very simple spectral events in the presence of broadband noise. The second algorithm deals with signals characterised by the periodic repetition of simple elements. This is a very common pattern in animal vocalisations. Finally, the third algorithm is adapted to highly structured signals with little variability in the structure. Highly structured signals are common among

Figure 5.1: An autonomous recording station powered by solar panels is dispatched on a boat.

song birds.

Simple Events: The Eurasian Bittern

The Eurasian bittern is a rare and threatened bird species living in large reed beds. Their habitats are difficult to access and the most obvious indication of the presence of the Eurasian bittern is the booming vocalisation of the male. Acoustical monitoring allows for passive investigation of bittern activity.

The call of the Eurasian bittern is very simple. It is almost completely characterised by its center frequency of about 150Hz. Calls typically occur in call sequences with a characteristic repetition frequency. In low noise conditions, this call can be detected by finding energy peaks in a suitable frequency band. Figure 5.3 shows a spectrogram of the bittern call. Each call begins with a short part at a slightly higher frequency. This part, however, cannot be used for pattern recognition because it can no longer be detected reliably in the presence of noise or at larger distances from the animal.

In order to achieve better frequency resolution in low frequency bands, input signals are downsampled by a factor of eight prior to further analysis.

Let $S(\omega, t)$ be the windowed power spectrum of a downsampled input signal s. The energy weighted novelty $N_{\ell,h}[S](t)$ for a frequency range from ℓ to h at time t is defined by

$$N_{\ell,h}[S](t) := \sum_{\omega=\ell}^{h} S(\omega, t)(S(\omega, t) - S(\omega, t+1))^2.$$

As indicated above, the calls to be detected are indicated by peaks in the novelty curve

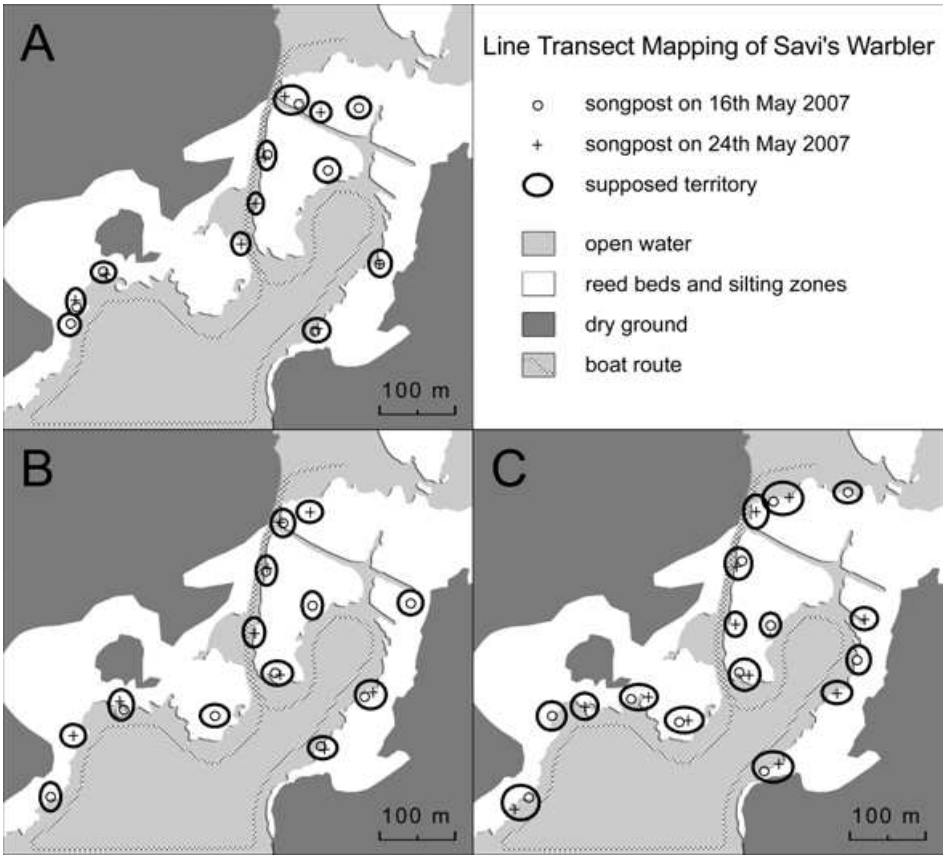


Figure 5.2: (Reproduced from [FTK08]) Estimation of breeding territories of Savi's warblers at Lake Parstein according to the criteria of line mapping. The position of the song posts were determined by A – mapping by an observer during the boat trip, B – listening to the recordings and determining the position of the boat on the base of the GPS track, C – recognising songs of Savi's warblers on the recordings using pattern recognition software and determining the position by GPS data.

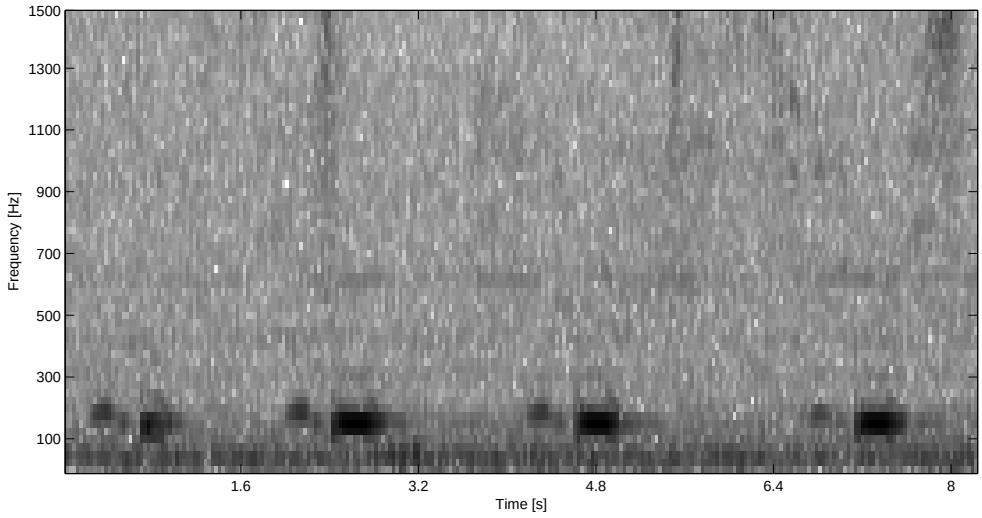


Figure 5.3: Spectrogram of a call sequence of the Eurasian bittern. The call is characterised mainly by high energy in a narrow frequency band at about 150Hz.

$N_{\ell,h}[S]$ for suitable values of ℓ and h. The main problem with this simple method, leading to false positive detections, is broadband noise overlapping the frequency band of the bittern call. This influence can be accounted for by estimating the noise level from a neighbouring frequency band. Figure 5.4 shows how broadband noise can be removed from the features by subtracting a low-pass filtered noise estimate.

From this, we derive a criterion $B[S](t)$ for the presence of bittern calls at time t as follows:

$$B[S](t) := N_{\ell_b,h_b}[S](t) - \alpha C_\phi[N_{\ell_n,h_n}[S]](t)$$

Here, $C_\phi[s]$ denotes convolution of a signal s with a given low-pass filter ϕ. This gives a smooth estimate of noise energy. For estimating the presence of the bittern call, bins ℓ_b to h_b of the spectrogram are examined. Similarly, noise is estimated from a neighbouring frequency band given by the bins from ℓ_n to h_n. A fixed factor α controls how much influence the noise measure has in the combined criterion.

Using the feature $B[S]$ directly for finding bittern calls still leads to a high number of false positive detections due to noise. We can, however, use the fact that the bittern usually calls in sequences with almost constant length pauses between calls. Figure 5.5 shows the features $B[S]$ for a 97 minutes recording from Lake Parstein. The recording is characterised by a high amount of noise caused by trains, wind, and water. Direct interpretation of the features would lead to a large number of false positive detections of the bittern call. An autocorrelation analysis of $B[S]$, however, allows to lower the number of false positives significantly. For this purpose, we calculate the windowed autocorrelation $A(\tau, t)$ of $B[S]$, where the time t gives the center of the window and τ describes the autocorrelation lag. From this, we derive a feature sequence

$$\tilde{A}(t) = \frac{1}{h}\left(\sum_{\tau=a}^{a+h-1} A(\tau,t)\right) - \frac{1}{k}\left(\sum_{\tau=b}^{b+k-1} A(\tau,t)\right).$$

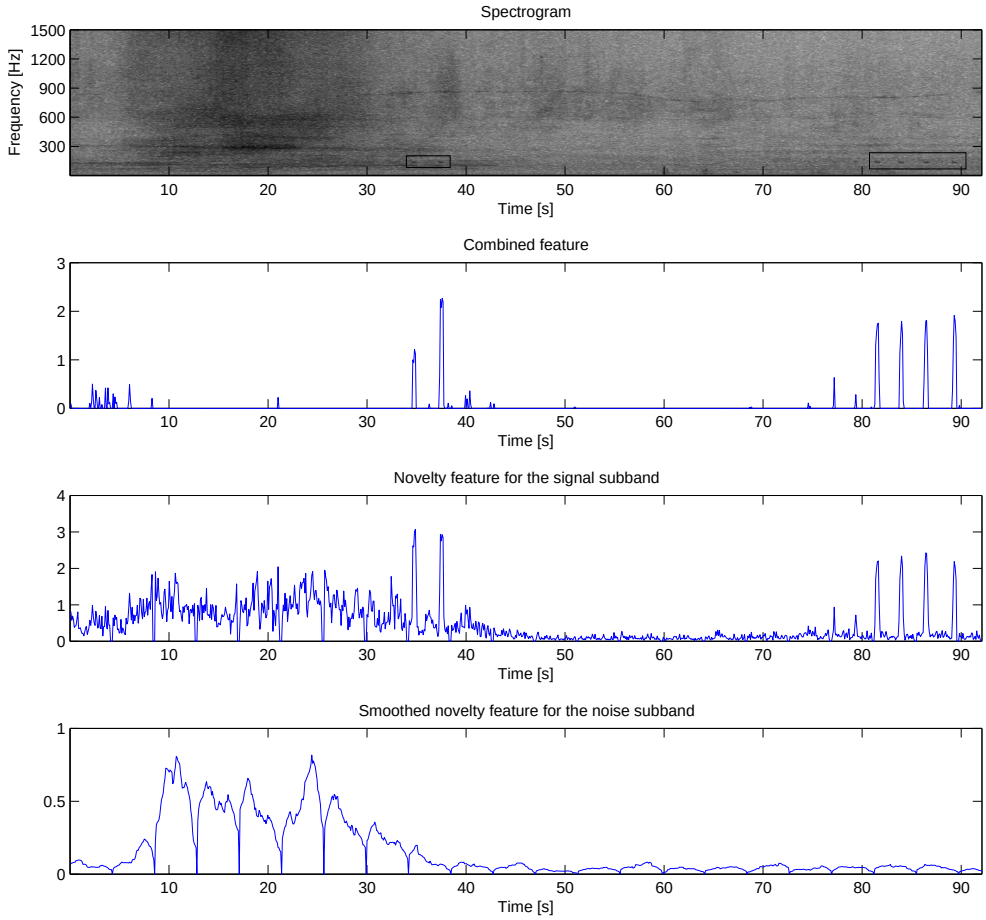


Figure 5.4: The impact of broadband noise on the features is removed by subtracting the weighted lowpass filtered feature from a neighbouring frequency band.

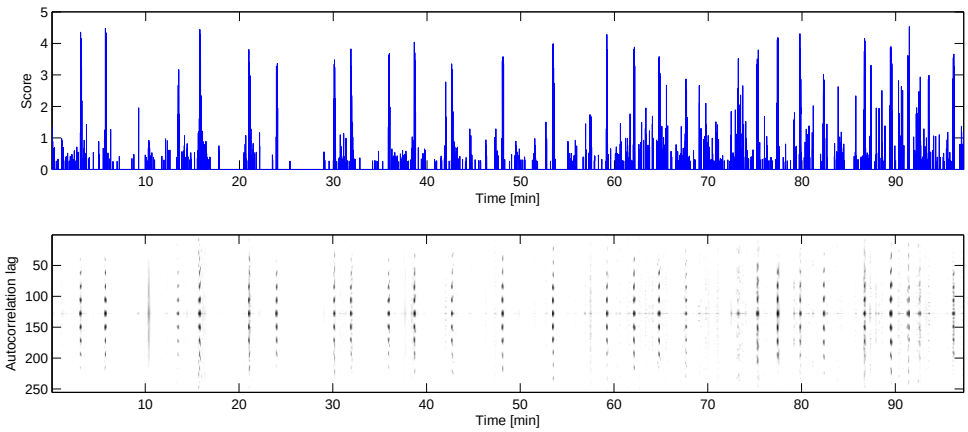


Figure 5.5: Features indicating activity in the frequency band characteristic for the Eurasian bittern after noise removal.

It measures the strength of the autocorrelation at lags $a \dots a + h - 1$ representing typical call repetition rates. We subtract the same measure for lags $b \dots b + k - 1$ indicating shorter repetition rates in order to remove the impact of noise events which show short repetition rates. Finally, candidate positions for the bittern call can be found from $\tilde{A}$ by peak picking combined with thresholding.

We have evaluated this algorithm for the detection of bittern calls using seven hours of recordings from Lake Parstein described in Section 5.2. The recordings are from five different days, some of them were taken from a boat, others from the shore. In Table 5.1, we give the number of time positions reported by the algorithm, the number of false positive detections (reported positions where no bittern activity is audible) and the number of false negative detections (positions, where bittern activity is audible, which were not reported by the algorithm).

The number of false positive detections is very much dependent on the acoustical situation. False positive rates were especially high on Day 3 and Day 5 for different reasons. On Day 3, a lot of speech is present in the recording. Especially in the beginning of the recording, a lengthy description of the recording equipment is spoken directly into the microphones and leads to a high number of false positives. On Day 5, the role of speech is taken by wind rhythmically impacting on the microphone. Most of the false positives can be explained by these two effects. Some of the false positives may also be effects due to the fact that the autocorrelation feature $\tilde{A}$ is computed from overlapping windows. This might result in finding one call sequence twice at different time positions.

False negatives are pleasantly seldom with our algorithm. They only occur in two situations: first, extremely silent calls are sometimes dismissed, especially when there are few consecutive calls. Second, our method for noise reduction in the features sometimes leads to a masking effect. When the energy in the band used for estimating noise levels is significantly higher than the energy in the band used for detecting the bittern call, the bittern call can be removed from the features although the call is clearly visible in the spectrogram. Such masking effects also occur in human hearing [ZF99] and can be the cause for calls which are undetectable by human listeners.

recording	duration [min:sec]	detections	false positives	false negatives
Day 1	15:21	3	0	0
Day 2	15:41	9	0	0
Day 3	116:51	74	20	7
Day 4	174:30	77	12	2
Day 5	97:15	83	52	1
Sum	419:38	246	84	10

Table 5.1: Results

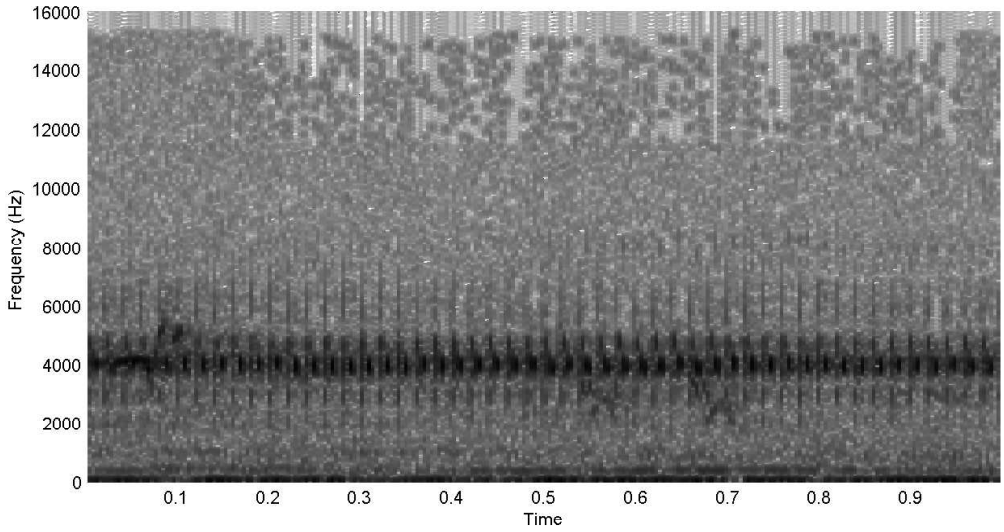


Figure 5.6: Spectrogram of the Savi's warbler's song.

Altogether, our algorithm is a very helpful tool in analysing the presence of the Eurasian bittern. Currently, the main problem is the high number of false positives from wind. How seriously this affects the utility of the algorithm depends on the task to be solved. For example, if the algorithm is used to demonstrate the presence of the Eurasian bittern in an area, most of the false positives can be disposed of by removing all reported positions with low values of $\tilde{A}$. This would result in dropping detections of short sequences of low calls of the bittern which is bearable as long as some longer or louder calls are present. If detecting very low calls is crucial then an additional feature discriminating wind from bittern calls is needed.

Element Repetition: Savi's Warbler

Another night active bird living in reed beds is Savi's Warbler. It has a very characteristic song formed by the continuous repetition of simple song elements at a rate of roughly 50 repetitions per seconds. An example of its song is given in Figure 5.6.

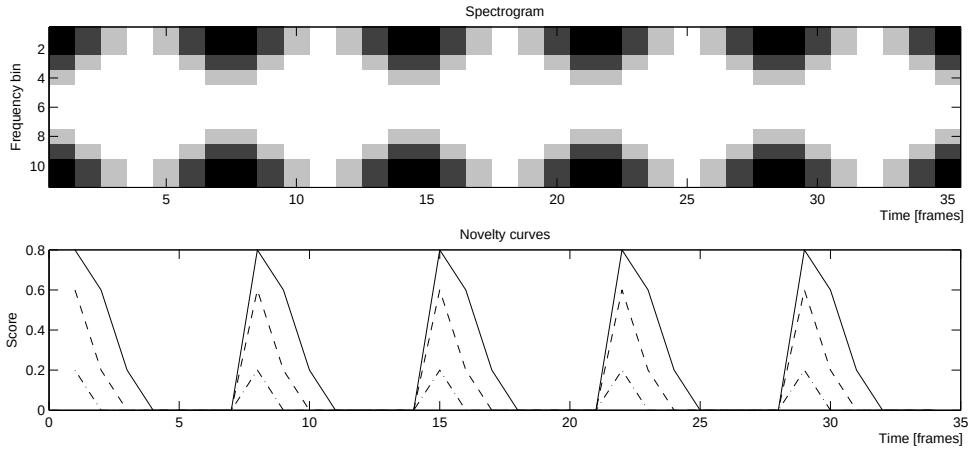


Figure 5.7: The ability of the novelty features to reflect song element onsets depends on the subband. The solid novelty curve is extracted from frequency bins 1 to 3, the dashed curve from bins 2 to 4 and the dot-dashed curve from bins 3 to 5.

The recognition of Savi's Warbler relies on detecting the presence of repeated elements with this typical repetition rate in the respective frequency band. The first step in estimating repetition rates is to compute a novelty curve $\tilde{N}$ similar to the one defined in the previous section. It is used to detect the onsets of song elements and is therefore designed to give peaks only when energy is rising in a given band. This version of the novelty curve is defined as:

$$\tilde{N}_{\ell,h}[S] := \sum_{\omega=\ell}^{h} \max\{S(\omega, t+1) - S(\omega, t), 0\}.$$

Now, repetition rates of song elements can be read off the autocorrelation function of the novelty curve. If the novelty curve has a period τ, its autocorrelation $A_{\tilde{N}}$ will show peaks at lags $n\tau$ for $n \in \mathbb{N}_0$. The presence of a period fitting to the repetition rate typical of the Savi's warbler's song is most easily detected via the Fourier transform of the autocorrelation function. Here, a strong peak in the frequency bin corresponding to the expected time lag indicates the right period.

How well song element onsets are reflected by the novelty feature is dependent on the frequency band that is used. Figure 5.7 shows an artificial example illustrating this effect. In some frequency bands, song elements may be well separated whereas in other frequency bands the elements overlap and thus do not lead to clear peaks in the novelty curve. Therefore, the frequency band known to contain the Savi's warbler's song is subdivided into five subbands. The novelty curve is computed for each subband and whenever the repetition rate we are looking for is detected in one of the subbands this subband will be selected for feature computation.

Similar to the strategy followed in the detection of the Eurasian bittern, noise reduction of the Fourier transform features described above can be conducted by subtracting the same type of features extracted from a flanking frequency band.

Finally the decision whether a Savi's warbler is singing at a given time is found by deciding whether its characteristic element repetition frequency is present for a long enough time. This

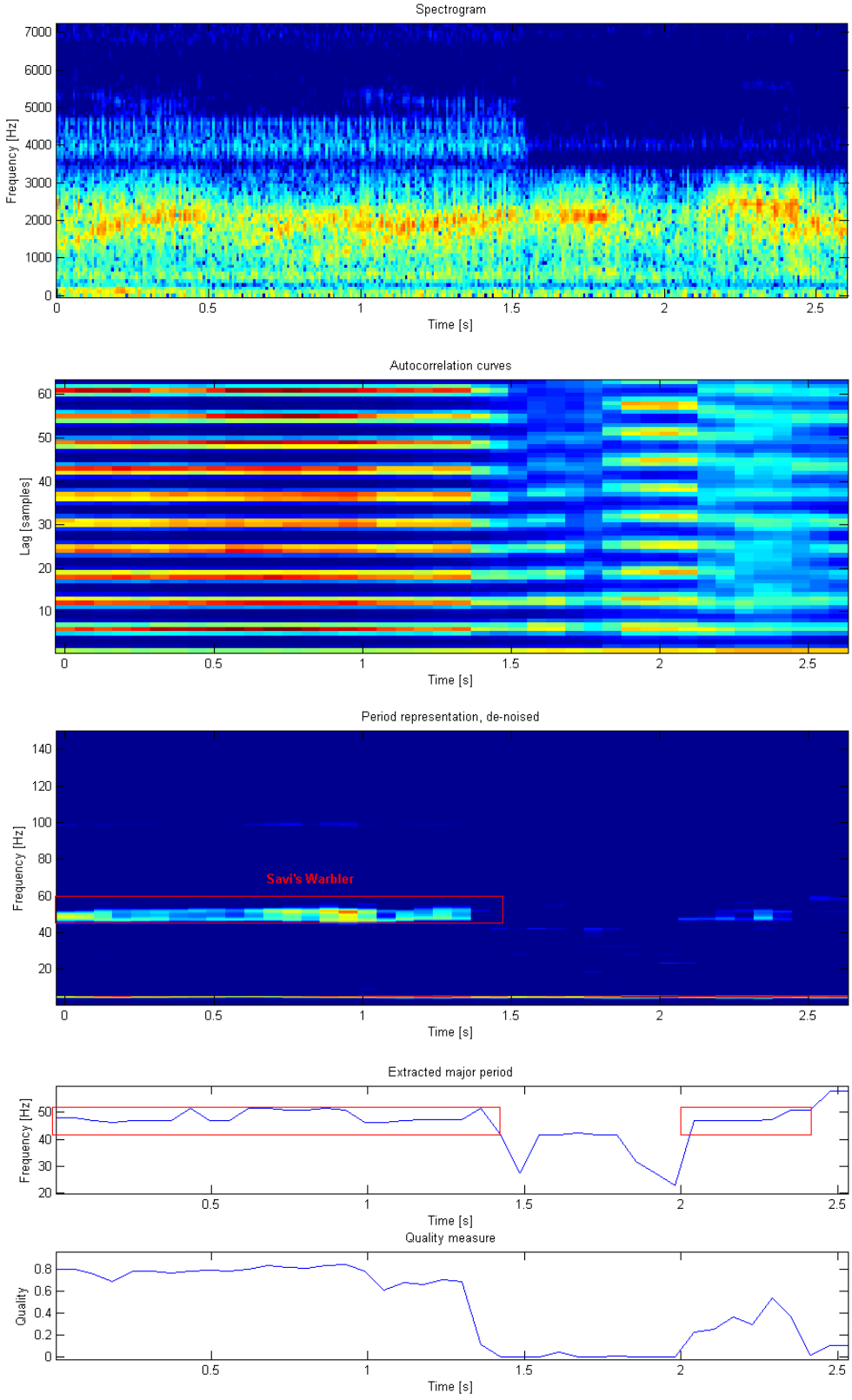


Figure 5.8: The presence of elements repeated with a given repetition rate is estimated using autocorrelation-based features.

is combined with a second criterion, the sharpness $\sigma[A_{\tilde{N}}]$ of the autocorrelation curve $A_{\tilde{N}}$ which is measured by the spectral flatness measure:

$$\sigma[A_{\tilde{N}}](t) := \frac{\prod_{\tau=1}^{a} \left(A_{\tilde{N}}(\tau,t)\right)^{\frac{1}{a}}}{\frac{1}{a}\sum_{\tau=1}^{a} A_{\tilde{N}}(\tau,t)}.$$

Here, $A_{\tilde{N}}(\tau,t)$ gives the windowed autocorrelation of the novelty curve $\tilde{N}$ at window t and time-lag τ. For each t the autocorrelation curve $A_{\tilde{N}}(\cdot,t)$ is assumed to be evaluated for a discrete time-lags.

An overview of the feature extraction process is given in Figure 5.8.

Evaluation of this algorithm has been performed on monitoring recordings from Lake Parstein. First results are available for 11 hours of audio material with frequent occurrence of the Savi's Warbler's song. The distance from the microphones is highly variable and the recordings contain a multitude of noise and background sounds. In order to reduce processing time, prior to the application of the detection algorithm, only those portions of the recording exceeding an energy threshold in the frequency band containing the Savi's warbler's song were analysed. This corresponds to the scenario where the presence of a bird in the recording area is to be verified. In this way, 106 minutes of the recording were selected and analysed. From these, 1339 seconds were correctly classified as song, 32 seconds of false positive detections were found and 587 seconds of the song were not detected. Moreover, we found that the detection of almost inaudible songs was possible.

A more detailed examination of this algorithm and a more detailed evaluation will be available in the thesis [Wol08] by D. Wolff.

Structure: The Chaffinch

The chaffinch is a very common and well-known bird. Many people can recognise its song in spite of its variability because it is strongly characterised by its structure. Figure 5.9 shows some of the variability found in the songs of chaffinches. Note, how the composition and form of the song elements vary strongly between songs. The general structure of the songs, however, is almost the same and can be used to describe an abstract model of the chaffinch song.

A chaffinch song typically consists of two to four segments in which one element is repeated, followed by an end segment. We therefore propose to detect chaffinch songs in three steps. First, all positions in a recording are found which are similar to typical end segments. Then, element repetition frequencies are estimated in a certain time window before the end segment. Finally, each candidate that has a combination of repeated element segments with parameters in a range typical of chaffinch songs is reported.

We will now describe the steps of this algorithm in more detail. Starting from a collection of 20 templates of typical end segments, we first find possible occurrences of the chaffinch song by dynamic time warping. Figure 5.10 shows a recording containing two chaffinch songs on the top and a template of an end segment on the left. In order to find potential occurrences of the template in the recording, a cost matrix C given by the difference of the template at position i and the recording at position j as matrix entry $C(i,j)$ is computed. If the template T is given by a sequence $(t_1,\ldots,t_m)$ of spectral vectors $t_i \in \mathbb{R}^d$ and the analysis signal S is given accordingly as a sequence $(s_1,\ldots,s_n)$ then the matrix $C \in \mathbb{R}^{m\times n}$ is given by:

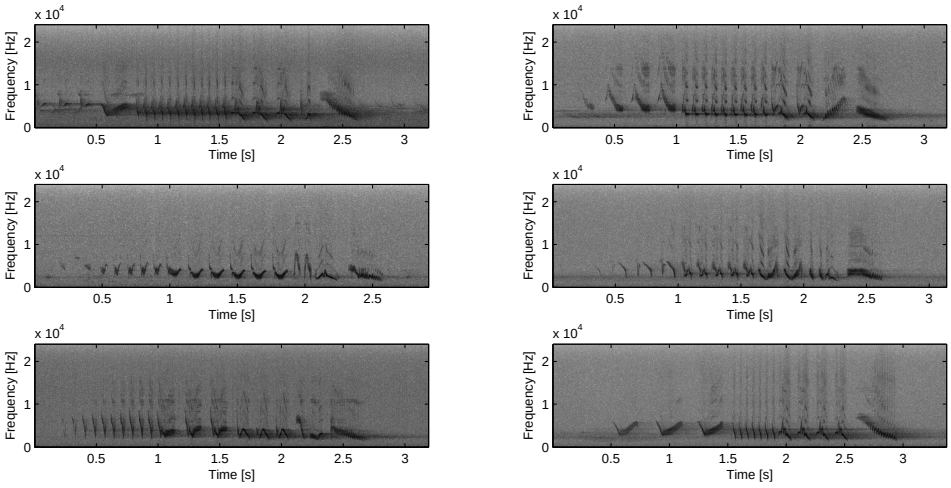


Figure 5.9: A collection of songs of the chaffinch. High variability in the form and composition of song elements is contrasted by the strong structural similarity.

$$C(i,j) = 1 - \frac{\langle t_i, s_j \rangle}{\|t_i\| \|s_j\|}.$$

Now, potential occurrences of the prototype can be found as paths through the matrix C giving a low sum of matrix entries. More precisely, such a path P of length k is given by a sequence $((p_{11}, p_{12}), \ldots, (p_{k1}, p_{k2}))$ where each entry (p_{i1}, p_{i2}) corresponds to matrix entry $C(p_{i1}, p_{i2})$ and thus assigns the vector $t_{p_i 1}$ of the template to the vector $s_{p_i 2}$ of the analysis signal. Two restrictions are applied to these paths in order to give sensible deformations of the template. First, each path has to start at the beginning of the template, i.e., $p_{11} = 1$ and must end at the last element of the template, i.e., $p_{k1} = m$. Moreover, paths should not walk backwards in time, remain stationary or move too fast. Therefore, the following restriction is enforced:

$$\forall i \in \{1, \ldots, k-1\} : (p_{(i+1)1}, p_{(i+1)2}) - (p_{i1}, p_{i2}) \in \{(0,1), (1,0), (1,1)\}.$$

Now, the starting positions in S of all paths P with total cost $\mathcal{C}(P) := \sum_{i=1}^{k} C(p_{i1}, p_{i2})$ below a given threshold are marked as candidates for end segments of the chaffinch song.

After the detection of potential end segments, each such candidate undergoes a second analysis step. In this step, we estimate the repetition rate of elements preceding the end segment. Figure 5.11 gives an overview of the features extracted for this reason.

Analogous to the algorithm for the recognition of Savi's warbler described above, computation starts by extracting the novelty curve $\tilde{N}$ from five subbands of the frequency range typical for chaffinch songs. Again, the novelty curves give a measure of change in the signal band over time. In particular, they show peaks where new elements begin. For each of these novelty curves, an autocorrelation curve is computed. At each time step, only one autocorrelation curve is chosen, which belongs to the subband giving the sharpest peaks in the autocorrelation curve. The resulting sequence of autocorrelation curves is called the adaptive

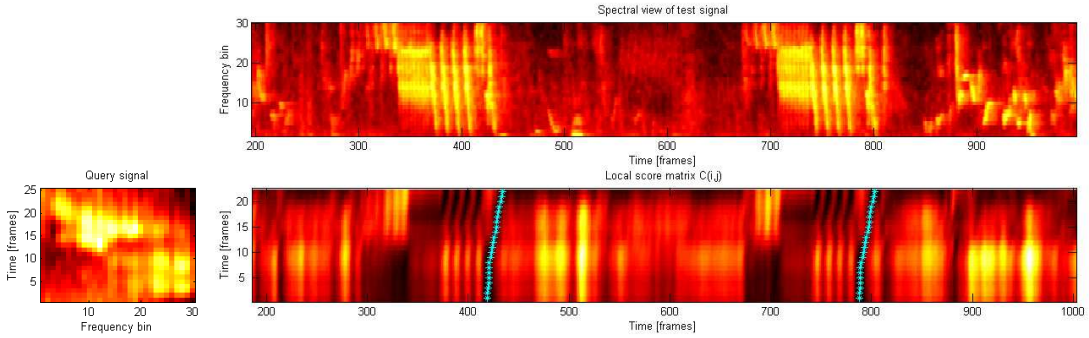


Figure 5.10: End segments of chaffinch songs are found by dynamic time warping.

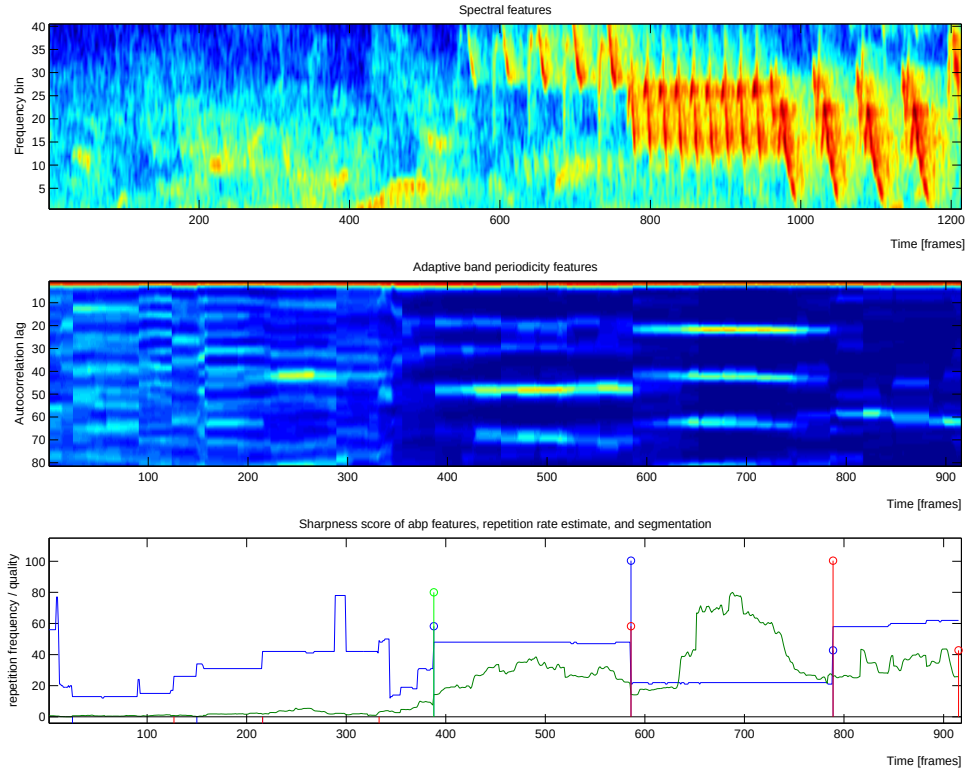


Figure 5.11: Repetition rate estimation for the part of the chaffinch song preceding the end segment.

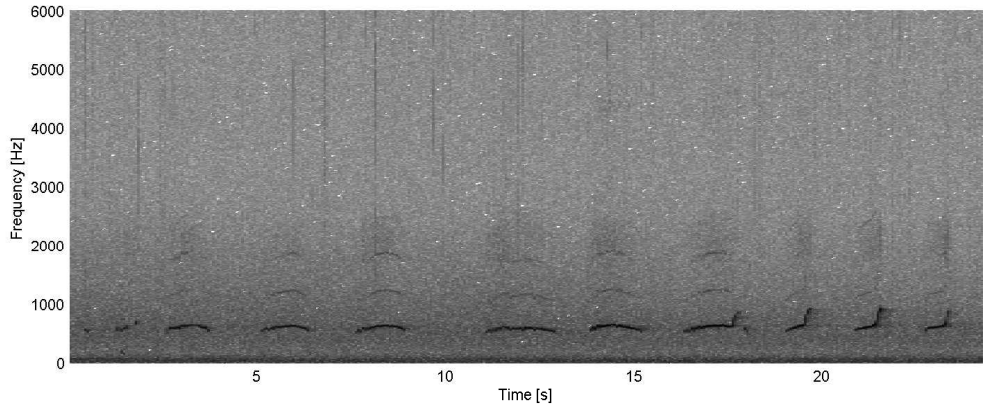


Figure 5.12: Spectrogram of the calls of the Eurasian tawny owl. Recognition of simple events may be useful as a first step for the recognition of these calls.

band periodicity (abp) features of the signal. The repetition frequency of elements is then indicated by the peaks of the autocorrelation curves.

Now, grouping these curves by repetition frequency leads to a segmentation of song candidates into segments of constant element repetition frequency. The final decision whether a song candidate is reported or not is based on the number and length of these segments and the repetition frequency of elements in each segment.

With this algorithm, chaffinch songs can be detected in fairly challenging signal-to-noise ratios. There are however certain cases when false negatives as well as false positives occur. By design, the algorithm cannot detect songs without an end segment or songs which are very short. The superposition of songs from other birds occasionally leads to false positive detections where one bird song resembles an end segment whereas the other consists of repeated elements.

Application to Other Species

In this section, we have described tools for the recognition of vocalisations of various bird species. They are not restricted to these species but should be applicable to similar cases.

First, the recognition of low complexity songs and calls such as that of the Eurasian bittern can be used for the recognition of other simple sounds in silent environments such as the songs of owls. It can also be the starting point of more complex recognition algorithms in the same way as the detection of end segments is a starting point for the detection of the chaffinch song. For example, Figure 5.12 shows the spectrogram of the call of the Eurasian tawny owl. It can be detected by looking for strong energy peaks at about 500Hz. In addition to that, it shows some harmonic structure which may be used to rule out false positives.

Second, there are a large number of animals whose vocalisations are characterised by the repetition of simple elements like in the song of Savi's Warbler. This suggests, that methods like those described above may not only allow recognising the vocalisations of different species of warblers but also of animals such as crickets, frogs and toads.

Finally, many song birds show a highly structured song like that of the chaffinch. Thus, the techniques above may be applicable to other birds like the blue tit, the coal tit or the wood warbler.

Chapter 6

Summary and Conclusion

6.1 Summary

In this thesis, we have examined the problem of algorithmic analysis of complex audio scenes with a special emphasis on natural audio scenes. One of the driving goals behind this work was to develop tools for monitoring the presence of animals in areas of interest based on their vocalisations. This task, which often occurs in the evaluation of nature conservation measures, leads to a number of subproblems in audio scene analysis, which have been addressed in this thesis.

As a first step in analysing natural audio scenes, a representative collection of the sounds to be expected in such scenes was necessary in order to find out their characteristics and variability. Building such collections is an enormous task beyond the capability of a single person. Therefore, we were lucky to find an extensive amount of digitised sound recordings of animals at the Animal Sound Archive of the Humboldt University of Berlin. Unfortunately, little infrastructure for searching and sharing this data has been available. In Chapter 2, we have therefore dealt with the problem of managing large collections of animal sounds. We have developed a distributed infrastructure for searching, sharing and annotating animal sound collections collaboratively. Some signal processing applications such as the automated detection of spoken comments in animal sound recordings have been developed.

Navigating such collections by metadata is very comfortable most of the time but not all sounds in the recordings are annotated. Moreover, finding similar sounds to a given recording can be very helpful in the study of animal vocalisations. Therefore, in Chapter 3, we have developed an index-based algorithm for similarity search in animal sound databases. Based on principles of image processing, we have designed features suitable for compact indexing of animal sounds and for their retrieval. In particular, we propose a method for similarity search in animal sound databases which is obtained by adding a novel ranking scheme to an existing inverted file based approach for multimedia retrieval.

In order to deal with the great complexity introduced by the high number of audio sources that are active simultaneously in natural audio scenes, we have studied source separation and array processing algorithms in Chapter 4. We have developed an algorithm to break down an audio scene into simpler components because previous algorithms trying to recover all sources of an audio scene exactly do not deliver results for natural audio scenes. For artificial mixtures of audio sources, we can extract one of the source signals almost exactly for mixtures of several audio sources and for mixture parameters varying over time. For natural audio scenes, we

achieve slight simplification compared to the input signals.

Finally, the problem of identifying animal species by their vocalisations has been studied in Chapter 5. Special purpose algorithms have been developed for the recognition of three species of birds. Two of these species are interesting for nature conservation purposes while the third is prototypical for song birds with highly structured songs. These algorithms have been evaluated on real world monitoring data and were found to be a helpful tool in this context, in particular, they are robust in adverse acoustical situations. Moreover, for one species, distribution maps of individuals could be generated automatically.

6.2 Perspective

The algorithms and infrastructure developed in this work can be seen as first steps on the way to automatised analysis and monitoring of natural and other complex audio scenes. But of course, there is still a lot of work that can be and has to be done in this field.

The information infrastructure described in Chapter 2 has now been available to the public in different states of development for several years. Most of the administrative tasks involved in managing the Animal Sound Archive are now carried out using its webinterface. A number of studies from bioacoustics to music research have been supported by the availability of this infrastructure. Moreover, it provides a good oppurtunity for the administration and distribution of annotated data sets for audio pattern recognition. It is a natural choice for hosting reference data sets in this field.

The similarity search algorithm of Chapter 3 has shown promising results for animal sounds showing trajectory-like spectral characteristics. Adding some features for the description of noise-like sounds should enhance it to become a useful tool in similarity search for the vocalisations of a large class of animal species. Being embedded in the Animal Sound Archive's information infrastructure it is hoped to show its utility for practitioners in bioacoustics.

The method for extracting low complexity audio components described in Chapter 4 is still in much of an experimental state and more work has to be done before it can evolve into a technique ready for application in the field. Nevertheless, it allows for the tracking of mixing parameters of an audio scene. It is thus conceivable to use this method for audio segmentation. Note, that whenever the component extraction algorithm is forced to choose strongly differing mixture parameters for adjacent time windows, a change in the mixing conditions of the audio scenes is indicated.

Problems abound in the context of animal species recognition based on sound. The special purpose detectors for interesting animal species described in Chapter 5 can be improved upon further. The results for the bittern detector show that wind and speech are the main reasons for false positive detections. Here, speech and wind detectors could provide useful preprocessors for all kinds of detectors. Furthermore, using learning algorithms for the classification of autocorrelation curves such as those used in the recognition of Savi's warbler might be a promising way of applying our methods to the recognition of further species.

More generally, the problem of pattern recognition in time-based data such as audio recordings is not very well understood. Algorithms in machine learning usually start from fixed dimensional representations of patterns. For these, a number of algorithms have been proposed which learn mappings from the fixed dimensional dataspace to class labels based on examples. In order to apply such methods to time-based data, fixed dimensional representations of it have to be found. It is not clear how this can be done in a systematic

way, especially where the signals involved are not locally stationary and thus are not well represented by hidden Markov models.

In order to stimulate discussion on the topic of bioacoustic pattern recognition, we have organised a number of national workshops. In an international workshop on the island Vilm in December 2007 [FBC08], we have brought together researchers from three continents. For the first time, experts from all over the world have come together for a focused discussion of computational bioacoustic methods for animal monitoring. The participants of this workshop are now forming a *worldwide network for computational bioacoustics for conservation*[1] which will hopefully lead to an increase of research in this field and a better communication of the possibilities pattern recognition can open up in nature conservation.

[1] A public mailing list is maintained for this network. Subscription is possible via `https://mailbox.iai.uni-bonn.de/mailman/listinfo.cgi/bioacoustic-monitoring`.

Bibliography

[AHFC01] E. Allamanche, J. Herre, B. Fröba, and Markus Cremer, *AudioID: Towards Content-Based Identification of Audio Material*, Proc. 110th AES Convention, Amsterdam, NL, 2001.

[Bar03] R. Bardeli, *Effiziente Algorithmen zur deformationstoleranten Suche in Audiodaten*, Master's thesis, University of Bonn, 2003.

[BCFK05a] R. Bardeli, M. Clausen, K.-H. Frommolt, and F. Kurth, *Ein verteiltes Informationssystem zur Forschungskooperation in der Bioakustik*, Fortschritte der Akustik, Tagungsband der DAGA, 2005.

[BCFK05b] ———, *Ein verteiltes Medienarchiv für bioakustische Datenbestände*, GI Jahrestagung (2), 2005, pp. 68–72.

[BM07] J. Brown and P. Miller, *Automatic classification of killer whale vocalizations using dynamic time warping*, Journal of the Acoustical Society of America **122** (2007), 1201–1207.

[BNF06] T. S. Brandes, P. Naskrecki, and H. K. Figueroa, *Using image processing to detect and classify narrow-band cricket and frog calls*, Journal of the Acoustical Society of America **120** (2006), 2950–2957.

[Car97] J.-F. Cardoso, *Infomax and maximum likelihood for source separation*, IEEE Letters on Signal Processing **4** (1997), 112–114.

[Cas01] M. Casey, *MPEG-7 sound-recognition tools*, IEEE Transactions on Circuits and Systems for Video Technology **11** (2001), 737–747.

[CBMN02] P. Cano, E. Batlle, H. Mayer, and H. Neuschmied, *Robust Sound Modeling for Sound Identification in Broadcast Audio*, Proc. 112th AES Convention, Munich, Germany, 2002.

[CKK03] M. Clausen, H. Körner, and F. Kurth, *An Efficient Indexing and Search Technique for Multimedia Databases*, SIGIR Workshop on Multimedia Retrieval, Toronto, Canada, 2003.

[Com94] P. Comon, *Independent component analysis - a new concept ?*, Signal Processing **36** (1994), 287–314.

[CSB87] J. H. Conway, N. J. A. Sloane, and E. Bannai, *Sphere-packings, lattices, and groups*, Springer-Verlag New York, Inc., New York, NY, USA, 1987.

[CTSO03] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, and M. Ouhyoung, *On Visual Similarity Based 3D Model Retrieval*, Computer Graphics Forum **22** (2003), no. 3, 223–232.

[DFS99] V. B. Deecke, J. K. B. Ford, and P. Spong, *Quantifying complex patterns of bioacoustic variation: use of a neural network to compare killer whale (orcinus orca) dialects*, Journal of the Acoustical Society of America **105** (1999), 2499–2507.

[DJLW08] R. Datta, D. Joshi, J. Li, and J. Z. Wang, *Image Retrieval: Ideas, Influences, and Trends of the New Age*, vol. 40, 2008, p. 60 pages.

[DMF+05] S. Derégnaucourt, P. P. Mitra, O. Fehér, C. Pytte, and O. Tchernichovski, *How sleep affects the developmental learning of bird song*, Nature **433** (2005), 710–716.

[DPH93] J. Deller, J. Proakis, and J. Hanson, *Discrete-time processing of speech signals*, Prentice Hall, New Jersey, 1993.

[Fag04] S. Fagerlund, *Automatic recognition of bird species by their sounds*, Master's thesis, Helsinki University of Technology, 2004.

[Fag07] ———, *Bird species recognition using support vector machines*, EURASIP Journal on Applied Signal Processing (EURASIP JASP) (2007), 8 pages.

[FBC08] K.-H. Frommolt, R. Bardeli, and M. Clausen (eds.), *Computational bioacoustics for assessing biodiversity. Proceedings of the International Expert meeting on IT-based detection of bioacoustical patterns*, BfN-Skripten, no. 234, 2008.

[FBKC06] K.-H. Frommolt, R. Bardeli, F. Kurth, and M. Clausen, *The animal sound archive at the Humboldt-University of Berlin: Current activities in conservation and improving access for bioacoustic research*, Advances in Bioacoustics 2, 2006, pp. 139–144.

[FC07] I. Farr and D. Chesmore, *Automated bioacoustic detection and identification of wood-boring insects for quarantine screening and insect ecology*, 4th International Conference on Bioacoustics, vol. 29, 2007, pp. 201–208.

[FG87] W. Förstner and E. Gülch, *A fast operator for detection and precise location of distinct points, corner and centres of circular features*, Proc. ISPRS Intercomission Conference on Fast Processing of Photogrammetric Data, June 1987, pp. 281–305.

[FH05] S. Fagerlund and A. Härmä, *Parametrization of inharmonic bird sounds for automatic recognition*, 13th European Signal Processing Conference (EUSIPCO 2005), 2005.

[FMK+03] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs, *A search engine for 3d models*, ACM Transactions on Graphics **22** (2003), no. 1, 83–105.

[FTK08] K.-H. Frommolt, K.-H. Tauchert, and M. Koch, *Advantages and Disadvantages of Acoustic Monitoring of Birds — Realistic Scenarios for Automated Bioacoustic Monitoring in a Densely Populated Region*, Computational bioacoustics for

assessing biodiversity. Proceedings of the International Expert meeting on IT-based detection of bioacoustical patterns. BfN-Skripten 234, 2008, pp. 83–92.

[GL03] G. Guo and S.Z. Li, *Content-based audio classification and retrieval by support vector machines*, IEEE Transactions on Neural Networks **14** (2003), 209–215.

[Här03] A. Härmä, *Automatic recognition of bird species based on sinusoidal modeling of syllables*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03), vol. 5, April 2003, pp. 545–548.

[HKO01] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*, John Wiley and Sons, 2001.

[HO97] A. Hyvärinen and E. Oja, *A fast fixed-point algorithm for independent component analysis*, Neural computation **9** (1997), no. 7, 1483–1492.

[HS04] A. Härmä and P. Somervuo, *Classification of the harmonic structure in bird vocalization*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004), 2004.

[HTR98] T. Hogg, H. Talhami, and D. Rees, *An improved synergetic algorithm for image classification*, Pattern Recognition **31** (1998), no. 12, 1893–1903.

[Hyv98] A. Hyvärinen, *New approximations of differential entropy for independent component analysis and projection pursuit*, Advances in Neural Information Processing Systems, vol. 10, 1998, pp. 273–279.

[KC76] C. H. Knapp and G. C. Carter, *The generalized correlation method for estimation of time delay*, IEEE Transactions on Audio, Speech, and Signal Processing **ASSP-24** (1976), no. 4, 320–327.

[KM98] J. A. Kogan and D. Margoliash, *Automated recognition of bird song elements from continuous recordings using dynamic time warping and hidden markov models: A comparative study*, Journal of the Acoustical Society of America **103** (1998), no. 4, 2185–2196.

[KM08] F. Kurth and M. Müller, *Efficient Index-Based Audio Matching*, IEEE Transactions on Audio, Speech, and Language Processing **16** (2008), no. 2, 382–395.

[LCL04] Q. Lv, M. Charikar, and K. Li, *Image similarity search with compact data structures*, CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management (New York, NY, USA), ACM, 2004, pp. 208–217.

[LOSX06] H. Lu, B. C. Ooi, H. T. Shen, and X. Xue, *Hierarchical indexing structure for efficient similarity search in video retrieval*, IEEE Transactions on Knowledge and Data Engineering **18** (2006), no. 11, 1544–1559.

[LZOS98] T-W. Lee, A. Ziehe, R. Orglmeister, and T.J. Sejnowski, *Combining time-delayed decorrelation and ICA: towards solving the cocktail party problem*, Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2, May 1998, pp. 1249–1252.

[MFW96] N. Mitsakakis, R. Fisher, and A. Walker, *Classification of whale song units using a self-organizing feature mapping algorithm*, Journal of the Acoustical Society of America **100** (1996), no. 4, 2644.

[MGGP06] N. J. Mitra, L. Guibas, J. Giesen, and M. Pauly, *Probabilistic fingerprints for shapes*, Symposium on Geometry Processing, 2006, pp. 121–130.

[Mil00] H. Mills, *Geographically distributed acoustical monitoring of migrating birds*, Journal of the Acoustical Society of America **108** (2000), no. 5, 2582.

[Mos01] A. Mosig, *Algorithmen und Datenstrukturen zur effizienten Konstellationssuche*, Master's thesis, University of Bonn, 2001.

[MSM+07] D. Mellinger, K. Stafford, S. Moore, R. Dziak, and H. Matsumoto, *An overview of fixed passive acoustic observation methods for cetaceans*, Oceanography **20** (2007), no. 4, 36–45.

[NBDS06] C. M. Nickerson, L. L. Bloomfield, M. R. W. Dawson, and C. B. Sturdy, *Artificial neural network discrimination of black-capped chickadee (poecile atricapillus) call notes*, Applied Acoustics **67** (2006), no. 11–12, 1111–1117.

[NK04] M. Novotni and R. Klein, *Shape Retrieval using 3D Zernike Descriptors*, Computer Aided Design 2004 **36** (2004), no. 11, 1047–1062.

[OBF04] M. Obrist, R. Boesch, and P. Flückiger, *Variability in echolocation: consequences, limits and options for automated field identification with a synergetic pattern recognition approach*, Mammalia **68** (2004), no. 4, 307–322.

[Pla99] J. C. Platt, *Fast training of support vector machines using sequential minimal optimization*, Advances in kernel methods: support vector learning (B. Schölkopf, C. Burges, and A. Smola, eds.), MIT Press, Cambridge, MA, USA, 1999, pp. 185–208.

[Plu05] M. D. Plumbley, *Geometrical methods for non-negative ica: Manifolds, lie groups and toral subalgebras*, Neurocomputing **67** (2005), 161–197.

[Pos02] G. Possart, *Signal classification of bird voices using multiscale methods and neural networks*, Master's thesis, University of Kaiserslautern, 2002.

[PSB+06] J. Placer, C. N. Slobodchikoff, J. Burns, J. Placer, and R. Middleton, *Using self-organizing maps to recognize acoustic units associated with information content in animal vocalizations*, Journal of the Acoustical Society of America **119** (2006), 3140–3146.

[Qui93] J. R. Quinlan, *C4.5: Programs for machine learning*, Morgan Kaufmann Publishers, 1993.

[RMV07] N. Rasiwasia, P. J. Moreno, and N. Vasconcelos, *Bridging the gap: Query by semantic example*, IEEE Transactions on Multimedia **9** (2007), no. 5, 923–938.

[Sch86] R.O. Schmidt, *Multiple emitter location and signal parameter estimation*, IEEE Trans. Antennas Propagation **AP-34** (1986), 276–280.

[SDK+03] F. Schwenker, C. Dietrich, H.A. Kestler, K. Riede, and G. Palm, *Radial basis function neural networks and temporal fusion for the classification of bio acoustic time series*, Neurocomputing **51** (2003), 265–275.

[Sel05] A. Selin, *Bird sound classification using wavelets*, Master's thesis, Tampere Universtity of Technology, 2005.

[SH03] P. Somervuo and A. Härmä, *Analyzing bird song syllables on the self-organizing map*, Workshop on Self-Organizing Maps (WSOM'03), September 2003.

[SH04] ———, *Bird song recognition based on syllable pair histograms*, Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2004), 2004.

[SHF06] P. Somervuo, A. Härmä, and S. Fagerlund, *Parametric representations of bird sounds for automatic species recognition*, IEEE Trans. Speech and Audio Processing **14** (2006), no. 6, 2252–2263.

[SK01] T. Seidl and H.-P. Kriegel, *Adaptable similarity search in large image databases*, State-of-the-Art in Content-Based Image and Video Retrieval (R. Veltkamp, H. Burkhardt, and H.-P. Kriegel, eds.), Kluwer Academic Publishers, 2001, pp. 297–317.

[SP03] P. Seekings and J. R. Potter, *Classification of marine acoustic signals using wavelets & neural networks*, Proceeding of 8th Western Pacific Acoustics conference (Wespac8), April 2003.

[STT07] A. Selin, J. Turunen, and J. T. Tanttu, *Wavelets in automatic recognition of bird sound*, EURASIP Journal on Signal Processing Special Issue on Multirate Systems and Applications **2007** (2007), 9 pages.

[Tay95] A. Taylor, *Recognising biological sounds using machine learning*, Proceedings of Eighth Australian Joint Conference on Artificial Intelligence, November 1995, pp. 209–212.

[Tho04] J. J. Thomson, *On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure*, Philosophical Magazine **7** (1904), no. 39, 237–265.

[TNH+00] O. Tchernichovski, F. Nottebohm, C. E. Ho, B. Pesaran, and P. P. Mitra, *A procedure for an automated measurement of song similarity*, Animal Behaviour **59** (2000), 1167–1176.

[TTSO06] J. T. Tanttu, J. Turunen, A. Selin, and M. Ojanen, *Automatic feature extraction and classification of crossbill (loxia spp.) flight calls*, Bioacoustics **15** (2006), no. 3, 251–269.

[Vap98] V. N. Vapnik, *Statistical learning theory*, Wiley-Interscience, 1998.

[VB88] B. Van Veen and K. Buckley, *Beamforming: A versatile approach to spatial filtering*, IEEE ASSP Magazine **5** (1988), 4–24.

[VBK01] R. C. Veltkamp, H. Burkhardt, and H.-P. Kriegel (eds.), *State-of-the-art in content-based image and video retrieval*, Kluwer Academic Publishers, Boston, Dordrecht, London, 2001.

[WBKW96] E. Wold, T. Blum, D. Keislar, and J. Wheaton, *Content-based classification, search, and retrieval of audio*, IEEE Multimedia **3** (1996), no. 3, 27–36.

[Wol08] D. Wolff, *Detecting Bird Songs via Periodic Structures: A Robust Pattern Recognition Approach to Unsupervised Animal Monitoring*, Master's thesis, University of Bonn, 2008.

[ZF99] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and Models*, Springer, Berlin, 1999.

[ZM06] J. Zobel and A. Moffat, *Inverted files for text search engines*, ACM Computing Surveys **38** (2006), 1–56.

[ZZS07] X. Zhou, X. Zhou, and H. T. Shen, *Efficient similarity search by summarization in large video database*, ADC '07: Proceedings of the eighteenth conference on Australasian database (Darlinghurst, Australia, Australia), Australian Computer Society, Inc., 2007, pp. 161–167.

List of Figures